CONTROL OF A LARGE SPACE STRUCTURE USING MULTIPLE MODEL ADAPTIVE ESTIMATION AND CONTROL TECHNIQUES

THESIS

Gordon Charles Griffin, Jr.
Captain, USAF

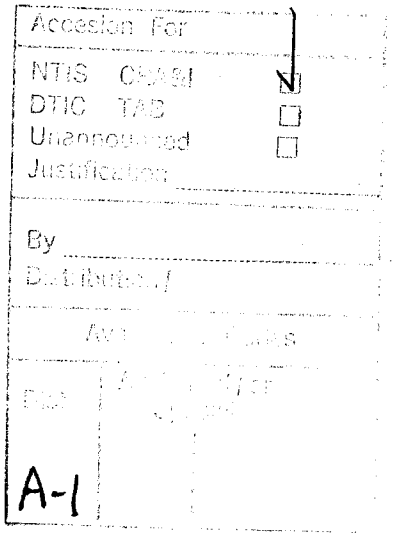
AFIT/GE/ENG/94D-14

DEPARTMENT OF THE AIR FORCE
AIR UNIVERSITY
AIR FORCE INSTITUTE OF TECHNOLOGY

Wright-Patterson Air Force Base, Ohio

AFIT/GE/ENG/94D-14

CONTROL OF A LARGE SPACE STRUCTURE USING MULTIPLE MODEL ADAPTIVE ESTIMATION AND CONTROL TECHNIQUES



THESIS

Gordon Charles Griffin, Jr.
Captain, USAF

AFIT/GE/ENG/94D-14

Approved for public release; distribution unlimited

AFIT/GE/ENG/94D-14

CONTROL OF A LARGE SPACE STRUCTURE USING MULTIPLE MODEL ADAPTIVE ESTIMATION AND CONTROL TECHNIQUES

THESIS

Presented to the Faculty of the Graduate School of Engineering
of the Air Force Institute of Technology
Air University
In Partial Fulfillment of the
Requirements for the Degree of
Master of Science in Electrical Engineering

Gordon Charles Griffin, Jr., B.S.E.E
Captain, USAF

December 1994

Preface

The purpose of this thesis is to apply moving-bank multiple model adaptive estimation and control (MMAE/MMAC) algorithms to the SPICE-4 space structure located at Phillips Laboratory, Kirtland AFB, NM. This research follows the work of Captain Greg Schiller who began the research with the SPICE-4 model. The primary design tools utilize Kalman filtering and LQG control. When uncertainties exist in the system model, a bank of filters increases the robustness of the LQG control. The moving bank limits the number of active filters in the parameter space, thus lessens the computational loading. The parameter space in this effort is three-dimensional, allowing for independent variation of the uncertain undamped natural frequencies of the bending modes of the SPICE-4 structure. Many techniques are investigated for moving the bank as well as for providing the control inputs. Results of this thesis indicate that the MMAE/MMAC algorithms are highly effective in quelling unwanted vibrations in the SPICE structure in the face of parameter variations.

There are many people without whom this thesis would not have been possible. First and foremost, I wish to thank my advisor, Dr. Peter Maybeck, for his guidance, suggestions, motivation and encouragement. I also wish to thank my committee members, Lt Colonel Riggins and Dr. Liebst, for their time and suggestions. I also wish to thank my fellow guidance and control students who have helped me through some rough times over the past few months. A special thank you goes to my predecessors, Capt Greg Schiller for sending disks, tapes, and other information from Kirtland AFB, and Capt Jim Fitch, for his help with the FORTRAN coding and in getting me started. Finally, I wish to thank my parents and brothers for their support, and my wife, Ann, for her patience, understanding, and love.

Table of Contents

page

page

page

List of Figures

page

page

page

page

page

List of Tables

page

Abstract

A moving bank multiple model adaptive estimator/controller (MMAE/MMAC) based on linear system, quadratic cost, and Gaussian noise (LQG) assumptions is used to quell unwanted vibrations in a large flexible space structure. The structure, known as the SPace Integrated Controls Experiment, or SPICE, exists at Phillips Laboratory, Kirtland AFB, NM. The structure consists of a large platform and a smaller platform connected by a tripod of flexible legs. The purpose of the control system is to maintain a very precise line-of-sight vector through the center of the spacecraft. Kalman filtering, used to estimate the position and velocity of the bending modes of the structure, and LQG control techniques are the primary design tools used in the MMAE/MMAC algorithms. Implementing a parallel bank of filters increases robustness when uncertainties exist in the system model, here specifically allowing adaptation to uncertain and changing undamped natural frequencies of the bending modes of the structure. A moving bank is utilized to reduce the computational loading. The MMAE/MMAC design provides an excellent method of estimating undamped natural frequency variations and quelling vibrations in the structure. The MMAE/MMAC was able to track numerous parameter changes and jumps while providing adequate control for the structure.

CONTROL OF A LARGE SPACE STRUCTURE USING MULTIPLE MODEL ADAPTIVE ESTIMATION AND CONTROL TECHNIQUES

I. Introduction

Kalman filtering has proven to be effective in estimating the states of a system with a known linear dynamics model driven by white, Gaussian noise. However, the dynamics model for a system is rarely known precisely. System parameters can vary over a range of values, typically due to temperature effects, aging or fatigue in the system, or even structural failure. For example, the uncertain parameters for a second order system might be the damping ratio and the undamped natural frequency. An optimal filter and controller is tuned to this set of parameters, yet, a small deviation from their nominal value may cause inappropriate or unstable control from a system based on a *nonadaptive* Kalman filter. Thus, a single Kalman filter/controller is often not robust enough to control the system adequately.

Multiple Model Adaptive Estimation (MMAE) is a technique that can overcome the effects of parameter uncertainty [21:131]. In the MMAE method, multiple Kalman filter models are generated, each based on a different assumed value of the uncertain parameters. Since the parameters typically vary over a range of continuous values, a finite set of parameter values must be selected as a result of discretizing the parameter space over the range of possible values. The multiple filters are then set up in a parallel bank. The state estimate of each filter is probabilistically weighted based on the probability that the filter-assumed parameter value is correct. The probability weights are determined by

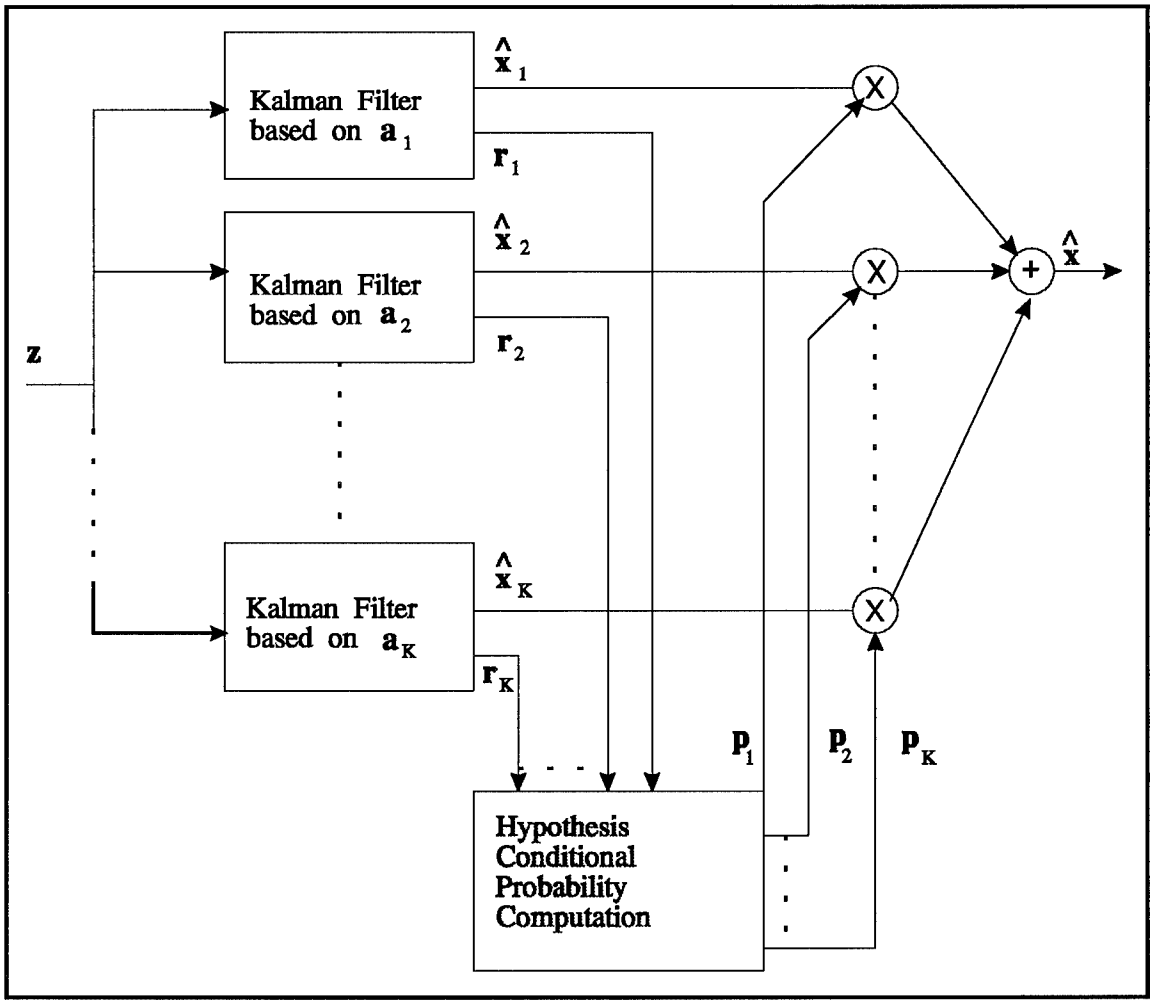


Figure 1-1. Multiple Model Adaptive Estimator [21:132]

the history of the measurement values and the residuals (the difference between actual measurements and best predictions of their values generated by the filter before the measurements are actually available) of each of the filters in the bank. Thus the filter based on the correct parameter value will have the highest associated probability. The sum of the individual probabilities will be equal to one, and the weighted outputs are summed to generate the overall adaptive state estimate. A diagram of the MMAE is presented in Figure 1-1. As illustrated, measurements of the outputs of the system,

$\mathbf{z}$, are fed into each Kalman filter (each based on the assumed parameter value, $\mathbf{a}_k$ for k=1,2,...,K) which then outputs a state estimate, $\hat{\mathbf{x}}_k$, and residual, $\mathbf{r}_k$. The residual $\mathbf{r}_k$ is the difference between the actual measurement, $\mathbf{z}$, and the Kalman filter's best prediction of the measurement, based on the assumption that the correct value of the parameter vector is $\mathbf{a}_k$. A probability weighting factor is generated for each filter state estimate as determined by a hypothesis conditional probability computation [21:130] applied to its residual and the residuals of the other filters. The probability weighting factors are multiplied by the appropriate state estimates. The MMAE state estimate is the resulting sum of all these products, which is a probability weighted average.

If the range of possible parameter values is very large, this technique would require a large number of Kalman filters, which would be computationally burdensome. By utilizing only a subset of Kalman filters in the parameter space, the computational load would be reduced. The concept of the moving bank involves using filters defined for only a subset of the full parameter space. At any one time only a small portion of the total number of filters will be on-line and actively estimating the system states. This subset of filters would attempt to surround the current parameter estimate in the parameter space, and is able to move around within the parameter space as the parameter values vary. This will be illustrated later in Section 1.2.3. This idea is based on the assumption that the probabilities associated with the filters that are closest to the actual parameter value should be predominant, and the probabilities of the filters not included in this subset should be negligible. Bank movement is accomplished by dynamically redeclaring which set of filters will be active, based on the properties of the residuals of the active filters. In this manner, the moving bank should be able to track the parameter estimate and thus maintain its validity.

The number of states for each Kalman filter must be reduced to aid further in reducing the computational burden. Tradeoffs between performance and computational loading must be analyzed when reducing the number of states of the system models. Also, proper tuning of the filters must be implemented to ensure adequate adaptation. Adding pseudonoise to the system models may compensate for the inadequacies in the reduced-order models, however, it may also inhibit the determination of which filter represents the true parameter value.

Once an adaptive state estimate is determined, it is used to generate the appropriate control to be input to the system. The controllers used in this research will be based on a Linear system model, with a Quadratic cost control criterion, driven by white Gaussian noises (LQG). A separate LQG controller will be developed and cascaded behind each Kalman filter in the MMAE. Thus, each controller output can be probabilistically weighted and then summed to form the optimal control to be input back into the system. The Multiple Model Adaptive Controller (MMAC) design is depicted in Figure 1-2. The LQG controller gain $-\mathbf{G}_c^*$ associated with each parameter value $\mathbf{a}_k$ is cascaded with the output state estimate from each associated filter and appropriately weighted (as discussed previously). The probability-weighted sum of all the weighted controls $\mathbf{u}_k$ from each individual LQG controller forms the final control which is input to the system. Additionally, each of the individual filters is provided knowledge of the final control as fed to the real world system (although this is not explicitly shown in Figure 1-2 in order to keep the diagram from becoming unnecessarily complicated).

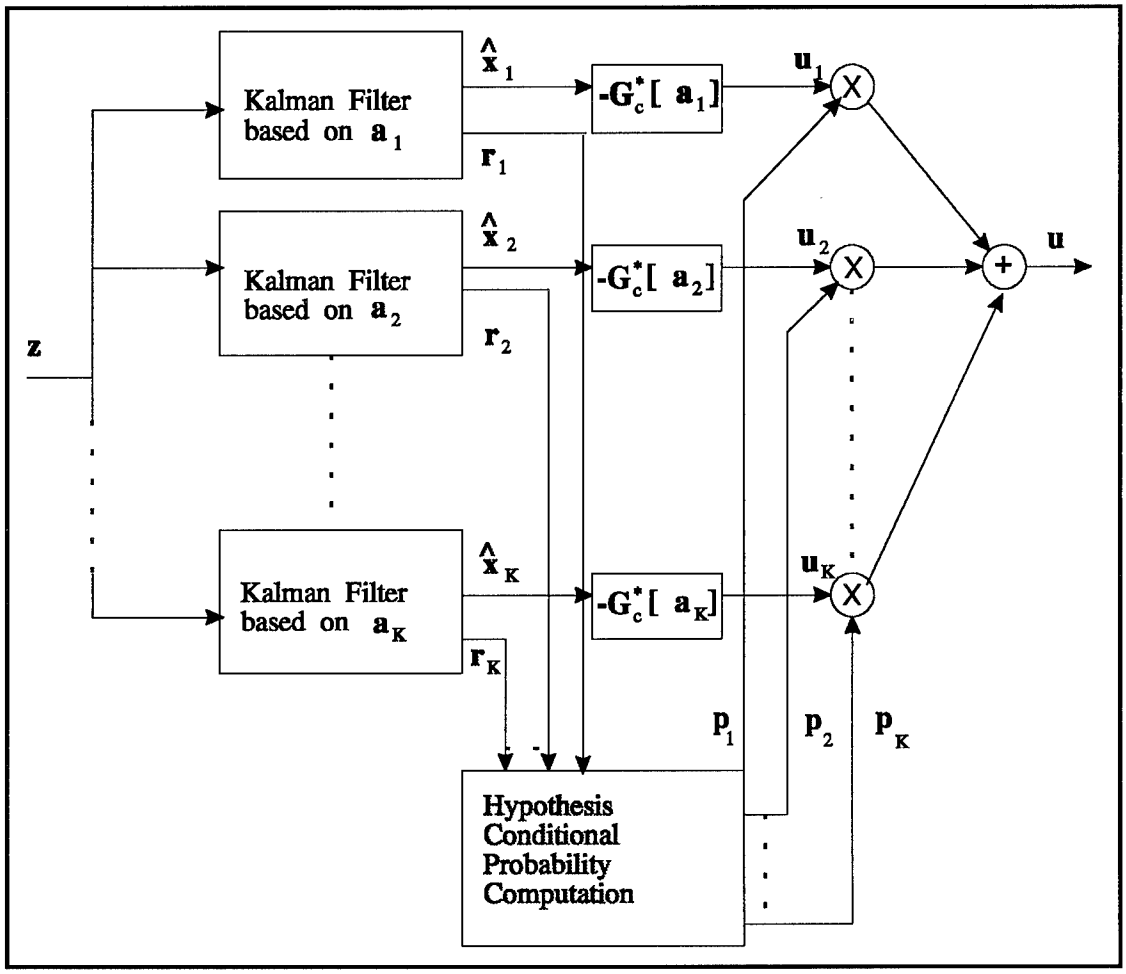


Figure 1-2. Multiple Model Adaptive Controller

This thesis will utilize a moving-bank MMAE and LQG-based MMAC controller to control a large flexible space structure (SPace Integrated Controls Experiment, or SPICE), with the primary goal being to quell unwanted vibrations induced in the structure. Figure 1-3 displays a simple rendering of the physical structure located at Phillips Laboratory, Kirtland AFB, NM. This work follows directly from work accomplished by Schiller [31]. In Schiller's work, the undamped natural frequencies were allowed to vary in concert with one another using a single multiplicative constant on all the undamped

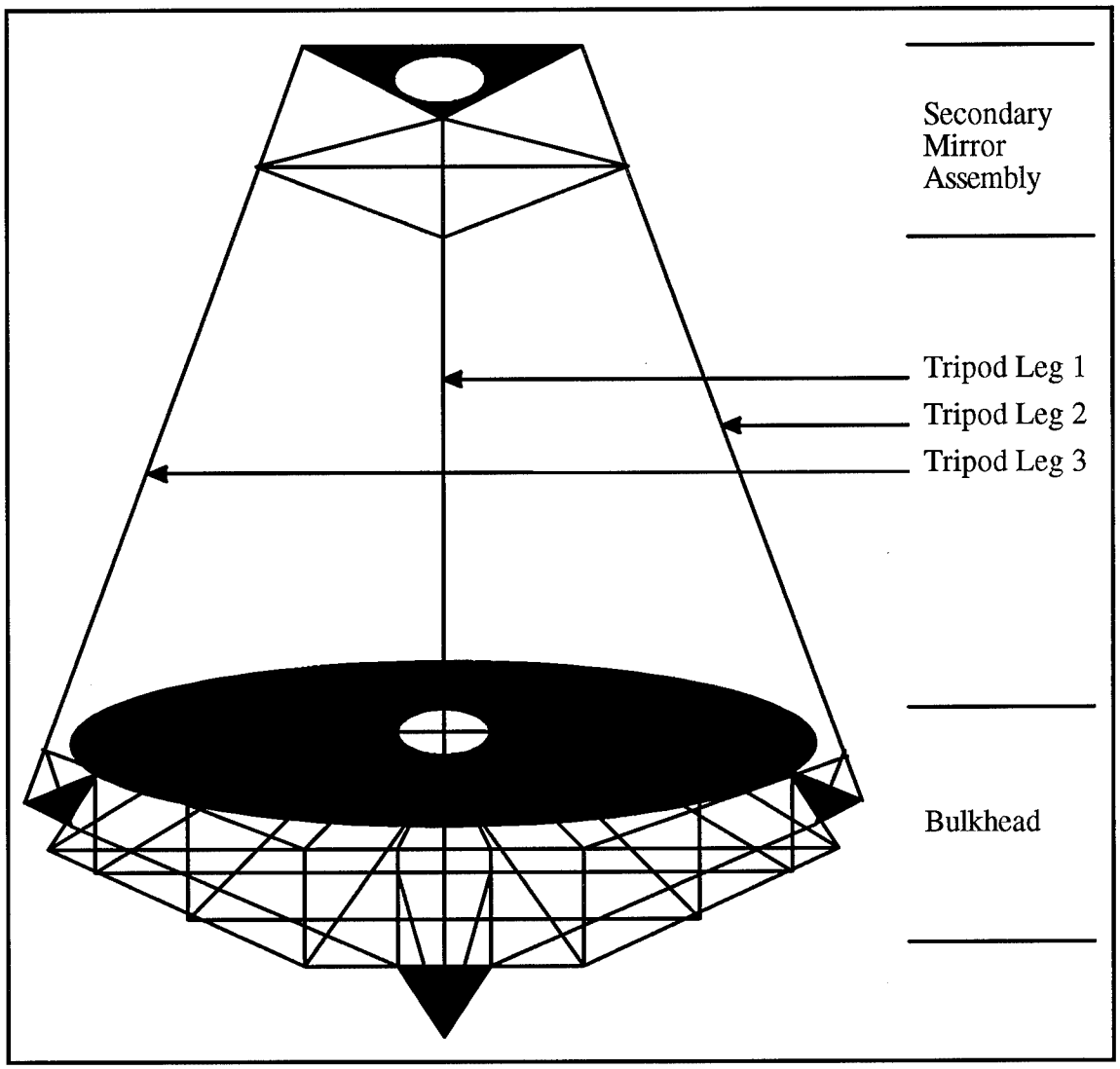


Figure 1-3: SPICE Space Structure

natural frequencies. In this work, the first three modes (corresponding to the lowest frequency modes in each of three directions: two translational and one torsional) will have one multiplicative constant, the second three modes will have another constant multiplier, and the rest of the modes will have yet another multiplier. This configuration is more realistic in that it allows for the independent variation of the undamped natural frequencies

of the system. Thus the parameter space for this effort becomes three-dimensional versus one-dimensional for Schiller's work.

1.1 Notation

Notation used in this thesis will attempt to maintain consistency with [20]. Deterministic and stochastic processes alike will be indicated by the roman typeface. Vectors will be displayed in bold-faced type, $\mathbf{x}$, whereas scalars will be normal type, x. Matrices will be displayed in upper case, $\mathbf{X}$. A particular realization of a variable will be displayed in italics, x.

1.2 Background

The purpose of this section is to introduce the four main areas of concern involved with this research. These four include: (1) the system model, (2) Multiple Model Adaptive Estimation (MMAE), (3) Moving-Bank MMAE, and (4) Moving-Bank Multiple Model Adaptive Controller (MMAC). The ideas will be presented here in a general sense. The specific problem for this research will be more fully developed in Chapters 2 and 3.

1.2.1 System Model

A full description of the SPICE-4 system model will be presented in Chapter 3. This system model is the same version that was used for Schiller's research [31]. Physically, the structure consists of a large platform which holds a deformable mirror assembly at the base, and a smaller platform and mirror assembly at the top. The two platforms are connected by a tripod of three legs. In the space environment, the three tripod legs will vibrate, causing the two mirror assemblies to be out of alignment, particularly when the vehicle is commanded to slew to accomplish pointing. This out-of-alignment condition is undesirable, thus the unwanted vibrations must be quelled. There are various accelerometers and actuators located throughout the structure. Successive

integrations of the outputs of the accelerometers measure velocity and position while the actuators generate the control inputs used to quell the vibrations of the structure. The control inputs are based on the accurate estimation of the undamped natural frequency of the bending modes of the structure.

The models for the SPICE-4 structure were developed at Phillips Laboratory by applying a finite element analysis in the determination of the mass and stiffness matrices. Chapter 3 will provide a discussion relating the mass and stiffness matrices to the damping ratios (ζ) and natural frequencies (ω_n) of all modeled bending modes. These last two parameters were initially considered to be unknown and formed the basis for the uncertain system model requiring numerous Kalman filters. However, Gustafson [7] showed that the damping ratio parameter does not have any appreciable effect on the adaptation process, thus it was not used for the work by Schiller [31] and will not be used for this research.

1.2.2 Multiple Model Adaptive Estimation

Multiple model adaptive estimation has proven to be effective in estimating the states for a system which contains uncertain parameters. If there are uncertain parameters, $\mathbf{a}$ (a vector composed of scalar parameters such as ζ and ω_n), the state model cannot be determined exactly. These parameters can be thought to fill a space defined by all the possible values the parameter vector may take ($\mathbf{a}_k$ for $k = 1, 2 \ldots K$). A separate Kalman filter must be generated for each parameter value in the space. The parameter space will most likely have a continuous range, thus, the continuous space must be purposefully discretized to keep the number of filters realizable. Since one filter is necessary for each discretized value, the total number of filters corresponds directly to the number of chosen values in the space.

As an example, let the two uncertain parameters of a system be the undamped natural frequency and the damping ratio, $\mathbf{a} = (\zeta, \omega_n)^T$ that are allowed to vary, each with 10 possible discrete values. Thus there are one hundred possible combinations of these parameters and the resulting parameter space is a two-dimensional 10x10 space, requiring 100 filters (K=100) as illustrated by Figure 1-4. If instead there were three uncertain parameters, the number of filters required jumps to 1,000. It is easy to see how the number of filters grows dramatically as the parameter space grows in size and as more uncertain parameters are added.

Within this full bank of Kalman filters, the output state estimate, $\hat{\mathbf{x}}_k$, from each filter is based on the parameter value, $\mathbf{a}_k$, used in its respective system model. The state estimate from the filter using the parameter value closest to the true value should be determined to be the most correct. The estimation algorithm uses information from the residual, $\mathbf{r}_k$, of each filter to determine which filter is most correct. The filter with the smallest residual (relative to the filter-computed residual standard deviation) is most likely

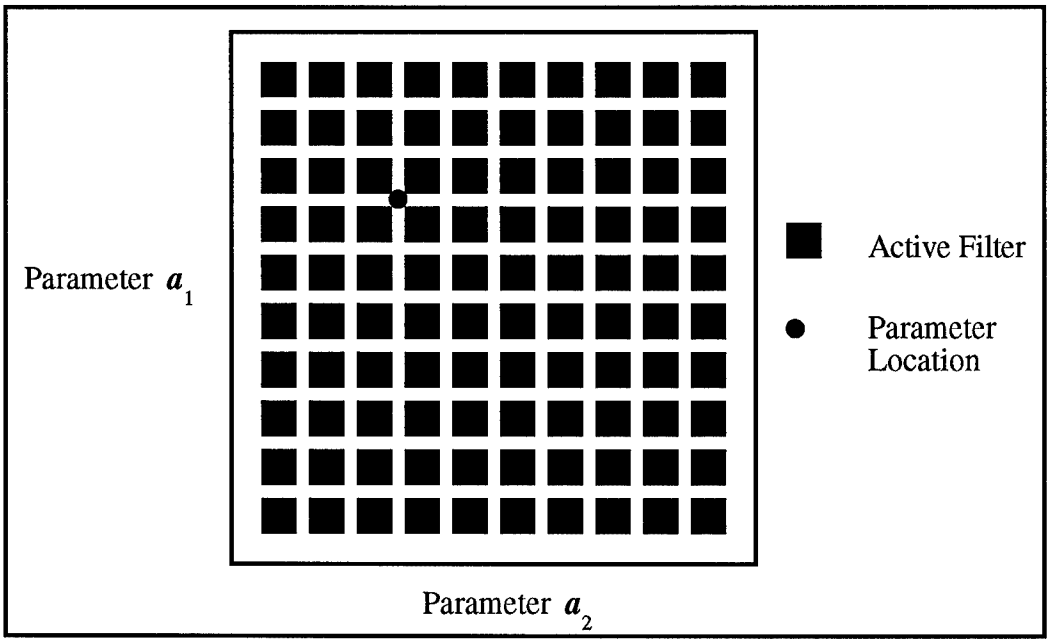


Figure 1-4. Full-Bank MMAE

the filter with the model based on the true parameter values, whereas the magnitudes of residuals from other filters should be relatively larger [21:133]. The residuals are used to produce a conditional probability to determine a weighting factor for the corresponding state estimate. This probability is the probability that the discrete parameter value used in the filter's system model is closest to the true parameter, conditioned on the history of measurements observed until the current time. Consequently, the highest conditional probability should be assigned to the most correct filter (smallest residual), and lower relative conditional probabilities assigned to the other filters [21:133]. There are four different methods for determining the final state estimate: (1) the Bayesian form (alluded to in Figure 1-1), (2) Maximum A Posteriori (MAP), (3) the Bayesian form with a Maximum Entropy with Identity Covariance (ME/I) density computation, and (4) the MAP form with the ME/I density computation. These methods will be presented in Chapter 2.

1.2.3 Moving-Bank MMAE

In problems in which there are a large number of discretized parameter points, and thus a large number of Kalman filters, there is a need to reduce the computational burden created by this large bank of filters in order to attain on-line applicability. The concept of the moving-bank MMAE was introduced in order to alleviate this problem. In the previous example, a 2-dimensional bank of 100 filters was introduced and is illustrated in Figure 1-4. It would be computationally infeasible to consider having all these filters active at once, thus, it is necessary to maintain a smaller subset of filters active at any time. Since the variable parameters (ζ and ω_n) can assume only one point or position in the two-dimensional parameter space at any time, a smaller bank of filters is used to estimate and surround the parameter value. For a two-dimensional parameter

space, nine active filters are typically used so that the actual parameter point would be surrounded by filters on all sides. The nine active filters are based on the fact that the filters based on parameter values nearest the actual parameter point would have the smallest residuals (best state estimates) and those based on values farther away would have larger residuals (relatively poorer state estimates). The filters based on parameter values that are farther away can be taken off-line without impacting the final state estimate appreciably, since their probability values, p_k, are negligible. Only the filters based upon closer point values would be active, which would result in the reduction of the computational load.

As the true parameters vary over time throughout the parameter space, the smaller subset of active filters must move with the true parameter. The computed estimate of the true parameter value must remain centered within the bank. As the parameters vary, the filters on the side farthest from the drifting point would be deactivated and ones nearest the new position would be activated. This smaller bank would then "move" by dynamic activation and deactivation of the filters, attempting to maintain itself centered about the true parameter point. This concept is depicted in Figure 1-5. In this case, the parameter position drifted diagonally, requiring that two sides be deactivated and redeclared. This figure also illustrates what will be referred to as a fine discretization, where the active filters' parameter points are adjacent to each other (at the finest level of discretization used for model generation).

A coarse discretization of the moving bank would be required in the initial parameter acquisition phase or if the true parameter value jumped by a large discrete amount. In these instances, the true parameter value is located outside the bounds of the

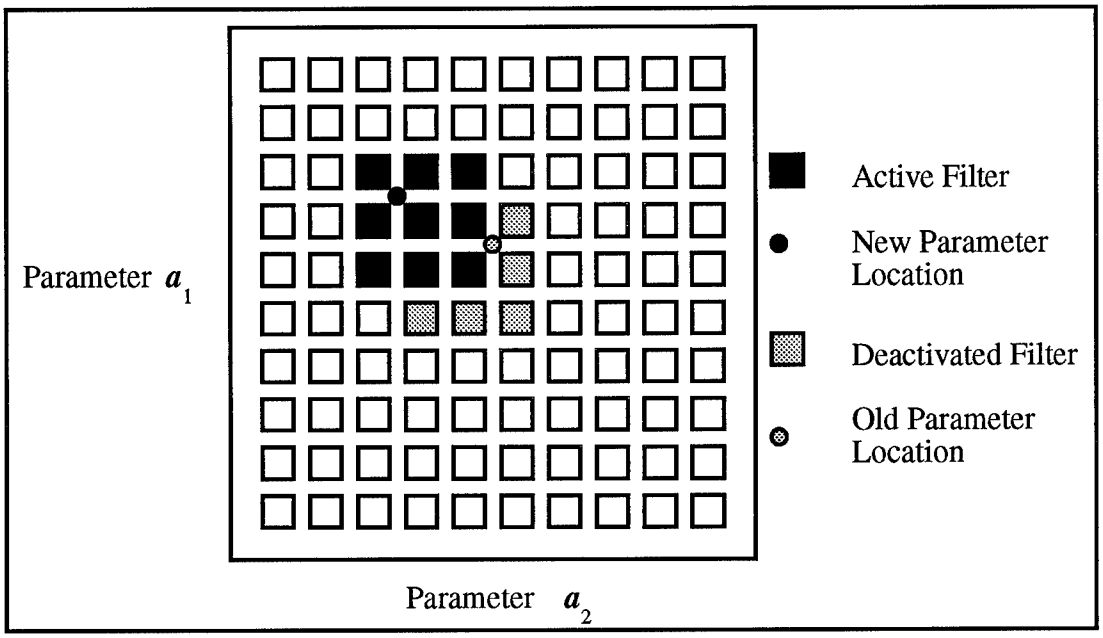


Figure 1-5. Moving Bank MMAE Fine Discretization

finely discretized bank, thus all filters in the bank will have large residuals. When this occurs, the bank becomes configured in a coarse discretization by activating the Kalman filters associated with the widest possible positions along the edges of the parameter space. This is illustrated in Figure 1-6. Once a parameter value is acquired, the bank can again be finely discretized to refine the estimate further.

In this research, three uncertain parameters will be addressed. As described previously, the first parameter will be the scalar multiplier on the undamped natural frequencies of the first three bending modes of the SPICE-4 system. The second parameter will be the multiplier on the second group of three undamped natural frequencies, and the third parameter will be the scalar multiplier on the remaining frequencies. The resulting parameter space becomes three dimensional.

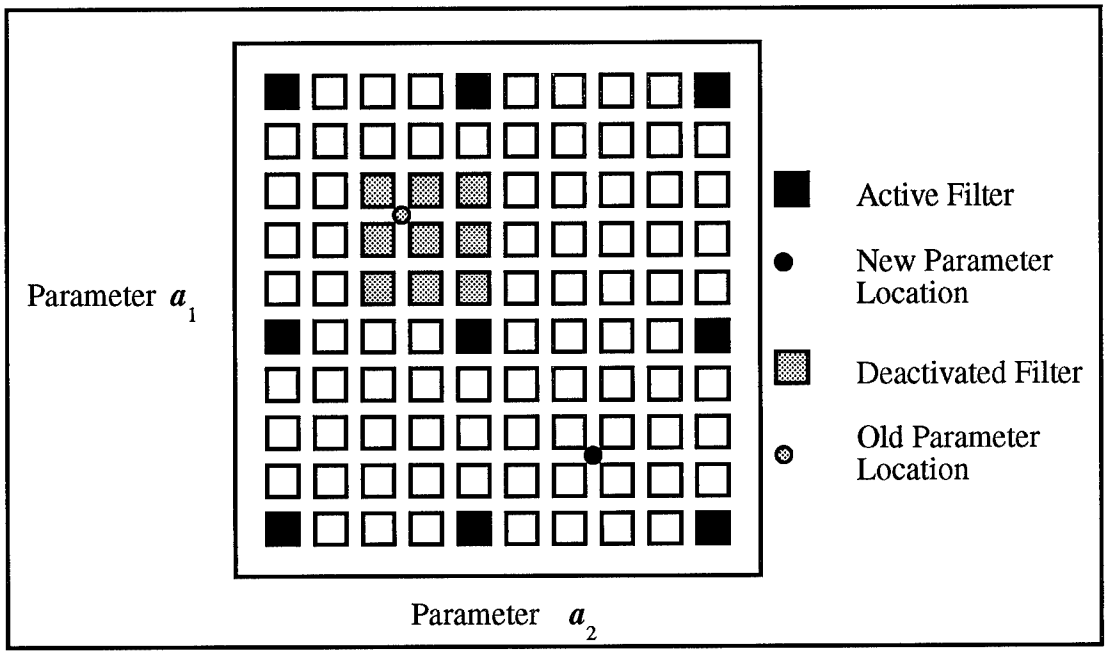


Figure 1-6. Moving Bank MMAE Coarse Discretization

There are 5 methods for deciding to expand, contract, move, or not move the bank. These decision making techniques, which will be discussed fully in Chapter 2, are: (1) Residual Monitoring, (2) Parameter Position Estimate Monitoring, (3) Parameter Position and "Velocity" Estimate Monitoring, (4) Probability Monitoring, and (5) Parameter Estimation Error Covariance Monitoring [26].

1.2.4 Moving-Bank MMAC

For the controllers used in this research, LQG regulators will be implemented since the desire to quell unwanted vibrations is equivalent to driving the system (bending mode) states to zero. Rigid body states are not included in the development of the controller since the purpose is to quell unwanted vibration in the structure, not the rigid-body motion. For this type of controller, the assumptions are that the system is adequately

modeled as linear, the cost is of a quadratic form, and the noise inputs are Gaussian [22:19].

The stochastic controllers developed for this research are based on the *certainty equivalence property* [22:17]. That is, the optimal controller gains are formed using the optimal LQ deterministic full-state feedback controller with the actual states of the system replaced by the state estimates from a Kalman filter. For the multiple model adaptive controller application, a separate LQG controller is formed from the state estimates of each Kalman filter in the bank as depicted in Figure 1-2. Using such a structure for a controller when there are uncertain parameter values is a form of "assumed certainty equivalence" [22:253]. Thus, each LQG controller is based on one discretized parameter position, as in the MMAE filters. A steady-state constant-gain control law can be used if the initial transients in the Kalman filter gains and the final transients in the optimal controller gains are assumed to be negligible. For this to be true, the control interval must be long compared to the two transient times.

There are many different ways for combining the MMAE and LQG controller. These are: (1) MMAC Control, (2) "Modified" MMAC Control, (3) MAP versus Bayesian MMAC Control, (4) Single Fixed-Gain Control, (5) Single Changeable-Gain Control, and (6) "Modified" Single Changeable-Gain Control. The methods differ in the way the state estimates are used or in the way the final control output is formed. The pure MMAC methods are referred to as Bayesian because their outputs are probabilistically weighted using a conditional probability calculation. The *maximum a posteriori*, or MAP method of MMAC does not form a probability weighted average; instead it simply utilizes the controller based on the single elemental filter with the highest probability calculation. All of these methods will be presented in more detail in Chapter 2.

1.3 Past Research

Research on the topic of moving bank multiple model adaptive estimation and control techniques has been ongoing at the Air Force Institute of Technology for many years. Research was initiated by Hentz in 1984 [9,26]. He was able to show that a moving-bank MMAE performed as well as a full-bank MMAE on a problem involving a single bending mode, but with a resulting order of magnitude less computational load [9]. Hentz began development of the algorithms and thresholds to move, expand and contract the bank, and began exploration of many of the controller design approaches discussed in the previous section.

Filios [2] used ambiguity function analysis in the performance evaluation and in determining the appropriate threshold levels for moving and contracting the bank. Unfortunately, he did not achieve very good results, partially due to the fact that the simple application he chose did not have a strong need for adaptive control [2:93].

Karnick [10,11] was the first to apply the techniques to a physical two-bay truss structure. The rectangularly shaped structure was designed to emulate an appendage, similar to a solar panel, which would then be attached to a much larger space structure. The entire truss and hub could be rotated in a planar motion, which resulted in undesired vibrations which needed to be quelled through adaptive control techniques. Karnick used a 6-state truth model and 6-state filter model. His results showed that the moving-bank MMAE was not able to identify the uncertain parameters, although it was sometimes able to provide accurate state estimates [10:93]. He found that the moving bank wandered throughout the parameter space and never provided a consistent parameter estimate. This problem was due to the sensors being too imprecise. Finally, due to the problems encountered, his results indicated that a coarsely discretized full-bank MMAE could perform as well as a finely discretized moving-bank MMAE [10:92].

Lashlee [13,14] continued the work of Karnick, focusing on the tuning aspects of the filter design. He found that, with a well-tuned filter, the uncertain parameters could be estimated accurately. He also showed that too much measurement noise can have an adverse effect on distinguishing between correct and incorrect models, and he demonstrated that the parameter space discretization process plays an important role in determining the parameter estimates accurately. As a result, Lashlee was able to show a greater performance potential for the moving-bank MMAE [13:199].

Van Der Werken [35] used the same two-bay truss as Karnick and Lashlee, but he expanded the truth model to 24 states. He conducted a performance analysis between the 24-state truth model and the 6-state filter model, and he examined the degradation caused by the order reduction between the two models. His results showed poor performance of the MMAE and MMAC for the reduced-order models. Thus, he concluded that the unmodeled states adversely impacted the ability of the moving bank to provide accurate parameter estimates [35:14]

Schore [27,32] continued Van Der Werken's work by examining some of the difficulties previously encountered. To determine which unmodeled states were causing the degradation, he compared the truth and filter model estimates of the true shape of the structure rather than its individual bending mode coordinates. He found that the reduced-order filter model should not degrade the MMAE and MMAC performance. After correcting some software problems, Schore demonstrated that the reduced-order MMAC controller was able to control the structure adequately despite unmodeled effects.

Moyle [29] concentrated on improving the estimator and controller algorithms, and further refined the Kalman filter and controller tuning parameters. These efforts resulted in significant improvements in performance. Moyle examined many different control techniques but had the best success with the Maximum Entropy with Identity

(ME/I) covariance, parameter position monitoring, and modified MMAC combination (all explained in Chapter 2) [29]. He also demonstrated that, contrary to the results obtained by Sheldon [33], packing the assumed parameter discrete values more densely at higher frequencies for the parameter discretization had no effect in improving controller performance [29].

Gustafson [7,8] was the first to implement MMAE and MMAC techniques to the SPICE space structure. He used the SPICE version 2 (SPICE-2) system model as the truth model, which consisted of 194 states. Gustafson developed reduced-order controllers using both internally balanced reduction and modal reduction; however, both types of reduced-order systems did not perform very well. His full order controller was able to provide stabilizing control over the range of parameter values, however, the control was not able to meet the RMS line of sight error specifications levied by Phillips Laboratory. Gustafson felt that the problem was not in the MMAC process, but was due to improper actuators and imprecise measurement devices on the structure [7]. Gustafson also determined that the controller was insensitive to damping ratio changes, and this parameter was deleted as an uncertain parameter without a loss of performance. Thus, his parameter discretization method resulted in a 1x7, one-dimensional parameter space, with the undamped natural frequency ω_n as the only changing parameter.

Fitch [3] continued work on the SPICE-2 model. He implemented a purposeful dither into the structure to enhance parameter identification. Sine wave, square wave, and wideband noise dither signals were all shown to enhance parameter identification significantly. He also more finely discretized the parameter space for the SPICE-2 model, and showed very good parameter tracking, even without dither enhancements [3].

Schiller [31] concentrated on applying a moving-bank MMAE/MMAC to the SPICE-4 structure. SPICE-4 was updated from SPICE-2 in that more test data was used

to validate the updated model for the structure, and the addition of actual sensors and actuators was implemented in the system models [16,31] . He developed several reduced-order models based on both the modal reduction and modal cost reduction techniques [31,34]. He discretized the parameter space more finely and over a larger total range than did Gustafson, resulting in 21 possible natural frequency values. Many different MMAE/MMAC implementations were tested, against multiple truth assumptions, with effective estimation and control provided by each technique.

1.4 Problem Statement

This thesis concentrates on applying a moving-bank MMAE/MMAC to the SPICE-4 space structure. The actual system models will be developed in Chapter 3. Previous work had always assumed that all of the undamped natural frequencies of the bending modes varied in concert with each other. That is, that there was a single scalar multiplier on all of the undamped natural frequencies [31]. For this thesis, the undamped natural frequencies will be partitioned into three groups. The first group will consist of the first three undamped natural frequencies of the bending modes of the structure, and will have one scalar multiplier. The second group will consist of the next three undamped natural frequencies, and will have another scalar multiplier. The remainder of the frequencies will have their own scalar multiplier. Thus the undamped natural frequencies of the three groups of bending modes will be allowed to move independently of each other. The three separate scalar multipliers will be the three uncertain parameters in the system model, so the resulting parameter space will be three-dimensional.

1.5 Scope

The focus of this research will be to apply moving bank MMAE/MMAC techniques to suppress unwanted vibrations in the SPICE-4 space structure. Since this research is a continuation of Schiller's work [31], the existing SPICE-4 models will be used, including the 294-state truth model and the various modal and modal cost reduced-order models developed by Schiller. The Kalman filter and controller tunings developed by Schiller will also be used for this effort, except where modifications are necessary. As explained above, the three uncertain parameters will allow the groups of undamped natural frequencies of the structural bending modes to move independently of each other. The discretized parameter space will be modified to allow a three dimensional full bank of size 7x3x7. (Early investigation showed that the parameter for the second group of frequencies did not require more than 3 discrete values. Thus, the moving bank, a 3x3x3 set of 27 filters, will only "move" in two of the three directions of parameter space.) Kalman filters and controllers will be developed, and the MMAE/MMAC algorithms modified to allow for the three dimensions. Performance evaluations will be conducted to show that the controller can adequately estimate the three uncertain parameters.

1.6 Approach

Since this effort is a continuation of Schiller's work, the truth model and the reduced-order models for the SPICE-4 system already exist and will be used. The Kalman filter and controller tunings developed last year will also be used for this effort. The development and testing of the three dimensional MMAE/MMAC for the total structure will consist of the following steps: (1) discretization of the parameter space, (2) development of the Bayesian MMAE, (3) development of the moving-bank algorithm, (4) development of the stochastic controller, and (5) performance evaluation.

Discretization of the parameter space. In Schiller's work, the parameter space was 1x21, with 21 scalar multipliers of natural frequency, ω_n. Additionally, this multiplier was applied to all of the flexible bending modes' frequencies simultaneously. For this research, it will be necessary to determine the size of the three-dimensional parameter space. The actual size of the complete parameter space is arbitrary, but based on physical insight and the need to keep the size reasonable, a parameter space of 7x3x7 was chosen. To accomplish the discretization, one parameter (scalar multiplier) is varied by discrete step sizes in one direction and successive Monte Carlo runs made until the Kalman filter becomes unstable (or until performance degrades unacceptably in some other respect). This procedure is repeated for each of the six directions (plus and minus in each dimension). Once a discretization level is determined for each direction, the remainder of the parameter space is completed by linearly changing the parameter values by the same amount until the parameter space is complete.

Bayesian MMAE development. Kalman filters for the truth and reduced-order model will be developed for each discretized parameter value in the parameter space. Both the ME/I and the ME/A methods for determining the probability weighting factors will be examined, and a lower bound for the probabilities will be implemented to avoid the "lock-out" condition. These ideas will be presented in detail in Section 2.3.2.

Development of the moving-bank algorithm. Algorithms from previous research will need to be changed to incorporate the increased dimension of the parameter space. Previous work indicates that the best method for the moving-bank algorithm is the parameter position method [31]. Residual monitoring and probability monitoring will also be reexamined. Methods and thresholds for contracting and expanding the bank will need to be examined due to the increased dimensions of the parameter space. Performance

analysis will be conducted for the cases of constant, slowly varying, and jump parameter changes.

Development of the stochastic controller. LQG techniques will be implemented in the design of the controller. Weighting matrix values developed by Schiller [31] will be used for this effort. Past research also indicated that the cross weighting matrix between states and controls could be set to zero with no appreciable impact [7]. The objectives of the controller are to achieve tight control of the bending modes without consistently saturating the actuators. Possible controller implementation techniques will be re-examined. These include MMAC, modified MMAC, MAP, and others, as discussed in Section 2.6.

Performance evaluation. Performance of the MMAE/MMAC will be evaluated via Monte Carlo analysis. The primary objective is for the tracking error to meet the specification of less than one μ-radian rms. The secondary objective is determining the accuracy of the state estimate, $\hat{\mathbf{x}}$, and the uncertain parameter estimate, $\hat{\mathbf{a}}$.

1.7 Summary

A general introduction to the concepts and operation of MMAE, moving-bank MMAE, and MMAC has been given in this chapter. A simple two-dimensional example demonstrated the basics of these concepts, yet the simple example can be easily extended to the complex SPICE-4 structure. A summary of the previous research on these topics has been presented. The chapter ended with a statement of the problem, the scope of the research, and the general methodology to be followed in attaining the solutions.

The remainder of this text is organized as follows. Chapter 2 will discuss the Kalman filter and LQG controller algorithms, as well as the MMAE/MMAC and the moving-bank decision making algorithms. Chapter 3 will introduce and develop the

system model for the actual total structure. Chapter 4 will discuss the software required and subsequent simulations necessary to investigate the system performance. Chapter 5 will present the results from this research, after which Chapter 6 will provide conclusions and recommendations for future research.

II. Background

2.1 Introduction

This chapter presents a basic development of the Kalman filter, the MMAE, the moving-bank MMAE, and the LQG controller algorithms. The various methods of implementing the MMAE and MMAC will be addressed. Also, a basic discussion of the mathematical modeling techniques, coordinate forms, and order reduction used in this thesis will be given. A more rigorous development of many of the subjects discussed in this chapter can be found [20,21,22,23,24]. A quick review of notation usage in Section 1.1 may be advantageous before reading this chapter.

2.2 Kalman Filter

A Kalman filter is used to estimate the uncertain states of a system. It is an "optimal recursive data processing algorithm" that can be shown to be optimal by essentially any standard [20], based on the assumptions that the system under consideration is adequately modeled as continuous and linear with white, Gaussian dynamics driving noise. Additionally, the measurement model is assumed to have linear sample-data measurements, corrupted by white, Gaussian discrete-time noise. The linear stochastic differential equation upon which the Kalman filter is based is:

$$\dot{\mathbf{x}}(t) = \mathbf{F}(t)\mathbf{x}(t) + \mathbf{B}(t)\mathbf{u}(t) + \mathbf{G}(t)\mathbf{w}(t) \tag{2.1}$$

where $\mathbf{x}(\cdot)$ represents an n-dimensional state vector, $\mathbf{u}(\cdot)$ is an r-dimensional deterministic control input vector, and $\mathbf{w}(\cdot)$ is an s-dimensional white Gaussian noise vector. $\mathbf{F}(\cdot)$ is an n-by-n dimensional system dynamics matrix, $\mathbf{B}(\cdot)$ is an n-by-r dimensional deterministic input matrix, and $\mathbf{G}(\cdot)$ is an n-by-s dimensional noise input matrix.

The white, Gaussian driving noise statistics are given by:

$$E\{\mathbf{w}(t)\} = \mathbf{0} \tag{2.2}$$

$$E\{\mathbf{w}(t)\mathbf{w}(t')^T\} = \mathbf{Q}(t)\delta(t-t') \tag{2.3}$$

where $\mathbf{Q}(t)$ is an s-by-s dimensional positive semi-definite matrix that indicates the strength of the dynamics driving noise. $\delta(\cdot)$ is the Dirac delta function.

Based on the assumptions stated previously, the state vector, $\mathbf{x}(\cdot)$, is a Gauss-Markov process and can be completely described by its mean, covariance, and covariance kernel. Initial conditions for both the mean and covariance must be determined to solve the above differential equation.. The initial mean, $\hat{\mathbf{x}}_0$, and covariance, $\mathbf{P}_0$, are given by:

$$E\{\mathbf{x}(t_0)\} = \hat{\mathbf{x}}_0 \tag{2.4}$$

$$E\{[\mathbf{x}(t_0) - \hat{\mathbf{x}}_0][\mathbf{x}(t_0) - \hat{\mathbf{x}}_0]^T\} = \mathbf{P}_0 \tag{2.5}$$

The initial mean is a best guess at the initial state vector, while the initial covariance describes a level of confidence in the estimate of $\hat{\mathbf{x}}_0$, or an indicator of the spread of the density function for possible initial values.

Since the Kalman filter algorithms will be implemented on a digital computer, a discretized version of the state space models is required. Instead of discretizing the filter equations, an equivalent discrete time system model should be developed from which a discrete-time filter will be built. This avoids using Riccati differential equation integrations and approximations [20:261]. The discretized system model upon which the Kalman filter is based is given by:

$$\mathbf{x}(t_i) = \mathbf{\Phi}(t_i, t_{i-1})\mathbf{x}(t_{i-1}) + \mathbf{B}_d(t_{i-1})\mathbf{u}(t_{i-1}) + \mathbf{G}_d(t_{i-1})\mathbf{w}_d(t_{i-1}) \tag{2.6}$$

where $\boldsymbol{\Phi}(\cdot,\cdot)$, the state transition matrix, is the solution to:

$$\dot{\boldsymbol{\Phi}}(t,t_{i-1}) = \mathbf{F}(t)\boldsymbol{\Phi}(t,t_{i-1}) \tag{2.7}$$

with the initial condition:

$$\boldsymbol{\Phi}(t_{i-1},t_{i-1}) = \mathbf{I} \tag{2.8}$$

Laplace transform techniques can be used to generate the solution based on a time invariant **F**-matrix as follows:

$$\boldsymbol{\Phi}(t_i,t_{i-1}) = \boldsymbol{\Phi}(t_i - t_{i-1}) = \mathcal{L}^{-1}\left\{\left[s\mathbf{I} - \mathbf{F}\right]^{-1}\right\}\Big|_{(t_i - t_{i-1})} \tag{2.9}$$

If $\mathbf{u}(\cdot)$ is held constant, as by a zero order hold, over one sample period, the deterministic matrix $\mathbf{B}_d(t_{i-1})$ of Equation (2.6) is given by:

$$\mathbf{B}_d(t_{i-1}) = \int_{t_{i-1}}^{t_i} \boldsymbol{\Phi}(t_i,\tau)\mathbf{B}(\tau)d\tau \tag{2.10}$$

The statistics of the discrete-time white Gaussian system dynamics driving noise are given by:

$$E\left\{\mathbf{w}_d(t_{i-1})\right\} = \mathbf{0} \tag{2.11}$$

$$E\left\{\mathbf{w}_d(t_{i-1})\mathbf{w}_d(t_j)^{\mathrm{T}}\right\} = \mathbf{Q}_d(t_{i-1})\delta_{(i-1)j} \tag{2.12}$$

where $\delta_{(i-1)j}$ is the Kronecker delta function. $\mathbf{Q}_d(\cdot)$ is given by:

$$\mathbf{Q}_d(t_{i-1}) = \int_{t_{i-1}}^{t_i} \boldsymbol{\Phi}(t_i,\tau)\mathbf{G}(\tau)\mathbf{Q}(\tau)\mathbf{G}^{\mathrm{T}}(\tau)\boldsymbol{\Phi}^{\mathrm{T}}(t_i,\tau)d\tau \tag{2.13}$$

where Equation (2.6) is modified such that $\mathbf{G}_d = \mathbf{I}$ if Equation (2.13) is used to evaluate $\mathbf{Q}_d$.

Sampled data measurements are obtained by the discrete-time measurement update model:

$$\mathbf{z}(t_i) = \mathbf{H}(t_i)\mathbf{x}(t_i) + \mathbf{v}(t_i) \tag{2.14}$$

where $\mathbf{z}(\cdot)$ is an m-dimensional discrete-time measurement vector, $\mathbf{H}(\cdot)$, is an m-by-n dimensional observation matrix, and $\mathbf{v}(\cdot)$ is an m-dimensional noise vector representing measurement noise. This noise vector is modeled by white, Gaussian noise with statistics given by:

$$E\{\mathbf{v}(t_i)\} = \mathbf{0} \tag{2.15}$$

$$E\{\mathbf{v}(t_i)\mathbf{v}(t_j)^{\mathrm{T}}\} = \mathbf{R}(t_i)\delta_{ij} \tag{2.16}$$

where $\mathbf{R}(t_i)$ is an m-by-m dimensional, positive definite symmetric matrix and δ_{ij} is the Kronecker delta function. The fact that $\mathbf{R}(t_i)$ is positive definite indicates that all measurements are corrupted with noise. Additionally, this model generation assumes that $\hat{\mathbf{x}}_0$, $\mathbf{w}(\cdot)$, and $\mathbf{v}(\cdot)$ are independent and thus, equivalently, uncorrelated due to the Gaussian assumptions [20:205].

After the discrete-time system models are developed, the Kalman filter proceeds through its estimation sequence which includes alternating propagation and update cycles. In the propagation segment, the state estimate after the most recent measurement update at time, t_{i-1}^{+}, is propagated forward to just prior to the next measurement update time, t_i^{-}. The superscript '-' denotes the state just prior to a measurement update whereas a '+' denotes the state after a measurement is incorporated. The propagation equations are given by:

$$\hat{\mathbf{x}}(t_i^{-}) = \mathbf{\Phi}(t_i, t_{i-1})\hat{\mathbf{x}}(t_{i-1}^{+}) + \mathbf{B}_d(t_{i-1})\mathbf{u}(t_{i-1}) \tag{2.17}$$

$$\mathbf{P}(t_i^{-}) = \mathbf{\Phi}(t_i, t_{i-1})\mathbf{P}(t_{i-1}^{+})\mathbf{\Phi}^{\mathrm{T}}(t_i, t_{i-1}) + \mathbf{G}_d(t_{i-1})\mathbf{Q}_d(t_{i-1})\mathbf{G}_d{}^{\mathrm{T}}(t_{i-1}) \tag{2.18}$$

When measurement data is received by the filter, the update segment consists of incorporating that data to form the best state estimate at time, t_i^{+}. The update equations are given by:

$$\mathbf{K}(t_i) = \mathbf{P}(t_i^{-})\mathbf{H}^{\mathrm{T}}(t_i)\left[\mathbf{H}(t_i)\mathbf{P}(t_i^{-})\mathbf{H}^{\mathrm{T}}(t_i) + \mathbf{R}(t_i)\right]^{-1} \tag{2.19}$$

$$\hat{\mathbf{x}}(t_i^+) = \hat{\mathbf{x}}(t_i^-) + \mathbf{K}(t_i)\left[\mathbf{z}(t_i) - \mathbf{H}(t_i)\hat{\mathbf{x}}(t_i^-)\right] \tag{2.20}$$

$$\mathbf{P}(t_i^+) = \mathbf{P}(t_i^-) - \mathbf{K}(t_i)\mathbf{H}(t_i)\mathbf{P}(t_i^-) \tag{2.21}$$

where $\mathbf{z}(\cdot)$ is the measurement to be incorporated at time t_i. The residual, or innovation, is defined as the difference between the current measurement and the Kalman filter's best estimate of that measurement prior to its occurrence, based on the entire time history of the measurements. The residual equation is given by:

$$\mathbf{r}(t_i) = \mathbf{z}(t_i) - \mathbf{H}(t_i)\hat{\mathbf{x}}(t_i^-) \tag{2.22}$$

where $\mathbf{r}(\cdot)$ is seen to equal the bracketed term in Equation (2.20) [20:228]. This residual, weighted by the Kalman filter gain, is used as an error correction factor to the previous state estimate, and is summed with the previous estimate to form an updated best estimate. The residual is known to be zero-mean, white, and Gaussian, with a covariance equal to the bracketed term in Equation (2.19) [20:229].

Assigning values to the initial state estimate, state covariance, dynamics noise strength, and measurement noise covariance is called tuning. The strength of the dynamic driving noise indicates how accurately the filter models the true dynamics of the system. Measurement noise indicates how much uncertainty is in the measurement values, due to either physical noise in the sensor or modeling imperfections. The filter should be tuned to attain the best performance in estimating the states of the system. To attain proper tuning, the filter-predicted state statistics should be roughly equal to the true statistics as output from the performance analysis based on the truth model.

Based on the assumption that the described system is time invariant with stationary noises, there are several simplifications that can be made to minimize the computational loading of implementing the filter. The typical Kalman filter response goes through an initial transient stage, followed by a steady-state stage [20:224]. If performance degradation is minimal, and if the initial transient stage is short compared to the total time

of interest, then a steady-state constant-gain Kalman filter can be implemented. In this case, the values for $\mathbf{K}(t_i)$, $\mathbf{P}(t_i^-)$, and $\mathbf{P}(t_i^+)$ in the previous equations can be taken to steady state and precomputed for all time. With the stationary noise assumption and a fixed sample rate, constant values for the $\mathbf{B}_d(t_i)$, and $\mathbf{Q}_d(t_i)$ matrices respectively need only be precomputed once. Additionally, the measurement noise covariance matrix, $\mathbf{R}(t_i)$, will be assumed constant.

The development of the specific SPICE-4 system model matrices and vectors, $\mathbf{x}(\cdot)$, $\mathbf{F}(\cdot)$, $\mathbf{\Phi}(\cdot,\cdot)$, $\mathbf{B}_d(\cdot)$, $\mathbf{u}(\cdot)$, $\mathbf{Q}_d(\cdot)$, $\mathbf{G}_d(\cdot)$, $\mathbf{z}(\cdot)$, $\mathbf{H}(\cdot)$, and $\mathbf{v}(\cdot)$ will be discussed in Chapter 3.

2.3 MMAE

The actual MMAE algorithm is based on the properties of Bayesian conditional probability densities. The basic concept of the MMAE was introduced in Chapter 1, but it will be amplified in this section. A more rigorous discussion of the subject can be found [21:129-139].

2.3.1 Bayesian Formulation

A Kalman filter is most effective in state estimation when the system model is precisely known. However, precise knowledge of the system model rarely occurs and uncertainties in these models may affect any or all of the system model matrices. In the SPICE-4 model, the uncertain parameter is assumed to affect the state transition matrix. This uncertain parameter, $\mathbf{a}$, may cause a subsequent loss of precision in the state estimation. As discussed in Chapter 1, $\mathbf{a}$ may be defined over a continuous range; thus an infinite number of Kalman filters would be required in the MMAE, each based on a specific realization $\mathbf{a} = \alpha$ in this range. Thus, the continuous range must be discretized such that the parameter $\mathbf{a}$ will take on a reasonable set of representative values $\mathbf{a}_1, \mathbf{a}_2, \ldots, \mathbf{a}_K$, where K is the total number of Kalman filters. For the example system described in

Chapter 1, the parameter space was 2-dimensional, with 10 discrete values for each scalar parameter, resulting in 100 Kalman filters. The actual discretization process will be discussed in Chapter 4.

Since the basic Kalman filter is developed from the Bayesian conditional probability density of the state, $\mathbf{x}$, conditioned on the measurement history, $\mathbf{Z}$ (where $\mathbf{Z}(t_i) = = [\mathbf{z}^T(t_1) \vdots \mathbf{z}^T(t_2) \vdots \cdots \vdots \mathbf{z}^T(t_i)]^T$ from Equation (2.14)), a Bayesian estimator in the case of uncertain states and parameters would add $\mathbf{a}$ to the left of the conditioning argument as follows, where the equality is derived from Bayes' rule:

$$f_{\mathbf{x}(t_i),\mathbf{a}|\mathbf{Z}(t_i)}(\xi,\alpha|\mathbf{Z}_i) = f_{\mathbf{x}(t_i)|\mathbf{a},\mathbf{Z}(t_i)}(\xi|\alpha,\mathbf{Z}_i) f_{\mathbf{a}|\mathbf{Z}(t_i)}(\alpha|\mathbf{Z}_i) \tag{2.23}$$

This equation is useful in that both quantities on the right hand side are known. The first quantity is the Gaussian density function totally defined by the outputs $\hat{\mathbf{x}}(t_i^+)$ and $\mathbf{P}(t_i^+)$ of the Kalman filter generated under the hypothesis that $\mathbf{a}$ assumes a specific realization α (i.e. $\mathbf{a} = \alpha$). The second term can be described by the following equation [21:131]:

$$f_{\mathbf{a}|\mathbf{Z}(t_i)}(\alpha|\mathbf{Z}_i) = \sum_{k=1}^{K} p_k(t_i)\delta(\alpha - \mathbf{a}_k) \tag{2.24}$$

where $p_k(t_i)$ is the hypothesis conditional probability (sometimes referred to as a weighting factor) that was introduced in Chapter 1, where $p_k(t_i) =$ $\text{prob}(\mathbf{a} = \mathbf{a}_k | \mathbf{Z}(t_i) = \mathbf{Z}_i)$. This probability is recursively determined by:

$$p_k(t_i) = \frac{f_{\mathbf{z}(t_i)|\mathbf{a},\mathbf{Z}(t_{i-1})}(\mathbf{z}_i|\,\mathbf{a}_k,\,\mathbf{Z}_{i-1})\,p_k(t_{i-1})}{\sum_{j=1}^{K} f_{\mathbf{z}(t_i)|\mathbf{a},\mathbf{Z}(t_{i-1})}(\mathbf{z}_i|\,\mathbf{a}_j,\,\mathbf{Z}_{i-1})\,p_j(t_{i-1})} \tag{2.25}$$

where the first term in the numerator is the probability density function of the current measurement, conditioned on the particular assumed parameter value and the observed past measurement history. This density function is evaluated as [21:132]:

$$f_{\mathbf{z}(t_i)|\mathbf{a},\mathbf{Z}(t_{i-1})}(\mathbf{z}_i|\mathbf{a}_k,\mathbf{Z}_{i-1}) = \frac{1}{(2\pi)^{\frac{m}{2}}|\mathbf{A}_k(t_i)|^{\frac{1}{2}}}\exp\{\cdot\}$$

$$\{\cdot\} = \{-\tfrac{1}{2}\mathbf{r}_k^{\mathrm{T}}(t_i)\mathbf{A}_k^{-1}(t_i)\mathbf{r}_k(t_i)\} \qquad (2.26)$$

where

$$\mathbf{r}_k(t_i) = [\mathbf{z}(t_i) - \mathbf{H}_k(t_i)\hat{\mathbf{x}}_k(t_i^-)]$$

and

$$\mathbf{A}_k(t_i) = [\mathbf{H}_k(t_i)\mathbf{P}_k(t_i^-)\mathbf{H}_k^{\mathrm{T}}(t_i) + \mathbf{R}_k(t_i)]$$

and $\hat{\mathbf{x}}_k(t_i^-)$, $\mathbf{P}_k(t_i^-)$, $\mathbf{H}_k(t_i)$, and $\mathbf{R}_k(t_i)$ are quantities in the *k*-th Kalman filter, which is based on the assumption that $\mathbf{a} = \mathbf{a}_k$, and m is the number of states in the state vector. Since the second term in the numerator of Equation (2.25) is the previous value of the hypothesis conditional probability, this equation constitutes an iterative solution. The denominator is the sum of the numerator terms from all *K* filters at time t_i. This can be considered a scaling factor that ensures the sum of all the individual conditional probabilities will equal one.

The resulting state estimate, conditioned on the entire measurement history, from the preceding development is given by [21:131]:

$$\hat{\mathbf{x}}(t_i^+) = E\{\mathbf{x}(t_i)|\mathbf{Z}(t_i) = \mathbf{Z}_i\}$$

$$= \int_{-\infty}^{\infty}\xi\left[\sum_{k=1}^{K} f_{\mathbf{x}(t_i)|\mathbf{a},\mathbf{Z}(t_i)}(\xi|\mathbf{a}_k,\mathbf{Z}_i)p_k(t_i)\right]d\xi$$

$$= \sum_{k=1}^{K}\hat{\mathbf{x}}_k(t_i^+)p_k(t_i) \qquad (2.27)$$

where $\hat{\mathbf{x}}_k(t_i^+)$ is the state estimate produced by the k-th Kalman filter, based on the parameter $\mathbf{a}_k$, and $p_k(t_i)$ is the probabilistic weighting factor, or the hypothesis conditional probability. The individual weighted state estimates are summed to create a weighted average final state estimate. This multiple model filter structure was depicted in Figure 1-1.

The conditional covariance development is similar in structure to Equation (2.27) as seen by the following equation [21:131]:

$$\mathbf{P}(t_i^+) = E\left\{\left[\mathbf{x}(t_i) - \hat{\mathbf{x}}(t_i^+)\right]\left[\mathbf{x}(t_i) - \hat{\mathbf{x}}(t_i^+)\right]^{\mathrm{T}} \middle| \mathbf{Z}(t_i) = \mathbf{Z}_i\right\}$$

$$= \sum_{k=1}^{K} p_k(t_i)\left\{\mathbf{P}_k(t_i^+) + \left[\hat{\mathbf{x}}_k(t_i^+) - \hat{\mathbf{x}}(t_i^+)\right]\left[\hat{\mathbf{x}}_k(t_i^+) - \hat{\mathbf{x}}(t_i^+)\right]^{\mathrm{T}}\right\} \quad (2.28)$$

where $\mathbf{P}_k(t_i^+)$ is the state error covariance from the Kalman filter based on the parameter value, $\mathbf{a}_k$. Unlike the individual $\mathbf{P}_k(t_i^+)$ values, the $\mathbf{P}(t_i^+)$ cannot be precomputed because it is dependent on the measurement history through $\hat{\mathbf{x}}_k(t_i^+)$, $\hat{\mathbf{x}}(t_i^+)$, and $p_k(t_i)$. Fortunately, this does not inhibit on-line application.

The statistics for the uncertain parameter $\mathbf{a}$ must be determined in order to evaluate the performance of the MMAE, but are not necessary in determining the state estimates. The conditional mean is given by the following [21:132]:

$$\hat{\mathbf{a}}(t_i) = E\left\{\mathbf{a}(t_i) \middle| \mathbf{Z}(t_i) = \mathbf{Z}_i\right\}$$

$$= \int_{-\infty}^{\infty} \alpha f_{\mathbf{a}|\mathbf{Z}(t_i)}(\alpha|\mathbf{Z}_i)\, d\alpha$$

$$= \sum_{k=1}^{K} \mathbf{a}_k p_k(t_i) \quad (2.29)$$

The corresponding covariance indicates the level of precision in the estimate of the parameter [21:133]:

$$\mathbf{P}_{\hat{\mathbf{a}}} = E\left\{\left[\mathbf{a}-\hat{\mathbf{a}}(t_i)\right]\left[\mathbf{a}-\hat{\mathbf{a}}(t_i)\right]^T \middle| \mathbf{Z}(t_i)=\mathbf{Z}_i\right\}$$

$$= \sum_{k=1}^{K}\left[\mathbf{a}_k-\hat{\mathbf{a}}(t_i)\right]\left[\mathbf{a}_k-\hat{\mathbf{a}}(t_i)\right]^T p_k(t_i) \tag{2.30}$$

2.3.2 Performance Evaluation and Enhancements

The preceding Bayesian multiple model adaptive estimator formulation has a few inherent problems. This section will address these problems specifically, discuss potential solutions, and introduce alternate methods for computing the final state estimate.

The first concern to be addressed relates to the relative sizes of the filter residuals. The residuals of the parallel bank of Kalman filters ultimately determine how well the MMAE will perform. The residuals of each of the elemental filters provide an indication of filter performance. The filter based on the most correct parameter value, $\mathbf{a}_k$, should have the smallest residual magnitude (scaled relative to the filter-computed residual covariance, $\mathbf{A}_k$). In this case, Equation (2.26) would provide the corresponding highest conditional density value and Equation (2.25) would subsequently provide the largest probability value for this "best" filter. However, if all the quadratic terms in the exponential portion of Equation (2.26) from each of the active filters are of the same relative magnitude, then the filter with the smallest $|\mathbf{A}_k(t_i)|$ term will have the largest probability assigned to it. This is not the proper criterion for determining probabilistic weights, thus this particular filter may not be appropriately weighted. Since the sum of all probability weights must be one, the remaining filters will also have erroneous weights, and the resulting final state estimate will be degraded.

To overcome this drawback of the straightforward Bayesian approach, the $\mathbf{A}_k(t_i)$ term in the conditional density function is replaced by an identity matrix which may be scaled by some constant. The resulting density function is thus given by:

$$f_{\mathbf{z}(t_i)|\mathbf{a},\mathbf{Z}(t_{i-1})}(\mathbf{z}_i|\mathbf{a}_k,\mathbf{Z}_{i-1}) = \frac{1}{(2\pi)^{\frac{m}{2}}|\mathbf{I}|^{\frac{1}{2}}}\exp\{\cdot\}$$

$$\{\cdot\} = \{-\tfrac{1}{2}\mathbf{r}_k^{\mathrm{T}}(t_i)\mathbf{I}\mathbf{r}_k(t_i)\} \qquad (2.31)$$

This method, termed the Maximum Entropy with Identity covariance (ME/I) method, attempts to account for the possibly erroneous affects of the $|\mathbf{A}_k(t_i)|$ term in the pre-multiplier and the exponentiated quadratic, as well as for an incorrectly evaluated $\mathbf{A}_k$ value due to model uncertainties. It assumes that the residuals follow a "maximally non-committed residual distribution" [33:24]. This method ensures that the residual with the lowest (absolute vs. relative) magnitude will be given the highest probability weighting factor.

Another proposal to remedying this situation is to rely solely on the exponential term in the conditional probability function (Equation (2.26)). This is accomplished by removing the term preceding the exponential in Equation (2.26). This removes the so-called "beta-dominance" effect of the $|\mathbf{A}_k(t_i)|$ in the premultiplier [28:473]. The result would no longer be a proper density function, however, due to the scaling effect in the denominator of Equation (2.25), the computed probabilities would still sum to one [28:473].

The second drawback to the Bayesian MMAE method lies in the iterative implementation of Equation (2.25). If the conditional probability of any elemental filter, $p_k(t_i)$, become zero at any time, it will remain at zero from that time forth. Thus, the MMAE would ignore that filter even if the elemental filter based parameter value, $\mathbf{a}_k$,

were to match the real value closely at a later time, because its contribution to the weighted average would be zero. This condition is termed "lockout". To prevent such a situation from occurring, a lower bound is artificially imposed upon the probability calculation. This eliminates the possibility of the probability ever going to zero. A threshold, p_{min}, is determined such that if the computed hypothesis conditional probability, $p_k(t_i)$, should fall below this lower bound, then $p_k(t_i)$ is set to a minimum value. The value is set so as not to bias the state estimate computation significantly, and the remaining probabilities are rescaled to maintain the summation equal to one.

Two methods can be applied in the evaluation of the final state estimate. The first method is the pure *Bayesian MMAE form* described in the preceding developments. The second method is the *Maximum A Posteriori* (MAP) method which makes the assumption that the filter state estimate with the highest probability p_k will be used solely as the final state estimate. This method does not provide for a "blending" of estimates from each of the elemental filters, as would the probability-weighted average estimate computations of Equations (2.27) and (2.29). Either of these two methods can be used with either the *Maximum Entropy with Identity covariance* (ME/I) density computation, or with the "beta-dominance" term stripped away as described above, as well as with the full computations of Equation (2.26).

2.4 Moving-Bank MMAE Development

In the full-bank MMAE developed above, one Kalman filter is required for each of K discrete parameter values. The computational load for all of these filters is quite demanding. A technique for alleviating this problem is to implement a moving bank of filters. The moving bank algorithm activates only a subset, J, of the total number of filters, K, where J<K at any one time. The moving bank will attempt to surround the parameter estimate, thus the bank must move, expand, and contract as the true parameter changes within the parameter space. Additionally, as the bank moves through the parameter space,

the Kalman filters need to be dynamically activated and deactivated. This section discusses the techniques for moving the bank, expanding the bank, contracting the bank, and initializing new filters.

2.4.1 Moving the Bank

The moving-bank MMAE is based upon the subset of J filters which surround the best estimate of the uncertain parameter, $\hat{\mathbf{a}}$. When a parameter variation occurs, the bank is moved in the direction of the new parameter estimate. This move can be accomplished in a finely discretized manner should the parameter drift slightly (as depicted previously in Figure 1-5) or in a coarsely discretized manner should the parameter make a large discrete jump (as depicted previously in Figure 1-6). There are 4 methods for implementing the move logic. These methods are: (1) Residual Monitoring, (2) Parameter Position Estimate Monitoring, (3) Parameter Position and "Velocity" Estimate Monitoring, and (4) Probability Monitoring [26]. It is necessary to determine the specific bank movement threshold values empirically for each method, which is analogous to filter "tuning".

2.4.1.1 Residual Monitoring

The residual monitoring method uses the likelihood quotient, $L_j(t_i)$, which is defined as:

$$L_j(t_i) = \mathbf{r}_j^{\mathrm{T}}(t_i)\mathbf{A}_j(t_i)^{-1}\mathbf{r}_j(t_i) \tag{2.32}$$

The likelihood quotient has the same form as the exponential term in Equation (2.26). The decision to move the bank is based on the value of this scalar quadratic function which is calculated for each of the J filters in the bank at each sample time. If the true parameter value lies outside the moving bank, large residual values will result for all the active filters, which will result in a large likelihood quotient for these filters. The smaller

the value of $L_j(t_i)$, the closer the filter-estimated parameter is to the true parameter. If all the filter's quotients are above a given threshold value, it indicates that the true parameter is no longer surrounded by the small moving bank. The bank is then moved in the direction of the filter with the smallest likelihood quotient. Past research has shown that determining the move threshold is critical for acceptable performance [9:61]. A threshold value that is too high results in move logic that is too slow in responding to parameter changes. A threshold value that is too low results in erratic movement of the bank throughout the parameter space, thus failing to maintain a proper estimate of the true parameter. Since $L_j(t_i)$ is a function of only the one most recent sample of measurement residual $\mathbf{r}_j(t_i)$, rather than a history of recent residuals, a single large measurement noise sample can adversely affect the performance of the residual monitoring method by causing unnecessary movement of the bank [26].

2.4.1.2 Parameter Position Estimate Monitoring

The parameter position estimate monitoring method attempts to keep the moving bank centered over the current estimate of the true parameter position determined by the MMAE, using Equation (2.29). The current estimate of the parameter's position is compared with the location of the center of the moving bank. If the difference is above a set threshold, then the bank is moved in order to re-center the bank over the currently estimated position. Unlike the previous method, this technique is not based solely on the most recent measurement value; it is based on the parameter estimate which is a function of the entire measurement history. As a result, this method does not suffer from the erratic behavior due to single large measurement noise samples [26]. When the true parameter moves to the edge or corner of the parameter space, the center of the bank cannot be centered on the parameter estimate without the bank exceeding the limits of the parameter space. Thus, the move logic must be altered slightly to account for this

possibility by placing the edge of the moving bank at the limits of the parameter space in these situations.

2.4.1.3 Parameter Position and "Velocity" Estimate Monitoring

Similar to the previous technique, this method attempts to estimate the position and "velocity" of the drifting parameter. The velocity estimate is then used to predict the location of the parameter position at the next measurement sample time. The velocity is determined by comparing the history of parameter position estimates over one sample period or several and dividing by the appropriate time increment. If the predicted position exceeds a threshold as compared to the current center of the moving bank, the bank is moved in the direction of the velocity vector. Previous research discovered that this technique resulted in no improvement in the estimation of the parameter, yet did exhibit a destabilizing effect on the bank's position [9:62]. Thus, this method will not be pursued in this effort.

2.4.1.4 Probability Monitoring

This method combines the properties of the residual monitoring method and the parameter position estimate monitoring methods in an attempt to center the bank over the estimated position based on the conditional probabilities as determined in Equation (2.25). The $p_k(t_i)$ with the highest value corresponds to the filter based on the parameter value closest to the true parameter value. If this probability is larger than a pre-determined threshold, the bank will move in the direction of the filter with the highest probability. As in position monitoring, this technique is dependent on the history of measurements and is not as susceptible to single large measurement noise samples as is the residual monitoring technique.

2.4.2 Expanding the Bank

When the true parameter position is not encompassed within the moving bank of filters, the bank has a more difficult time determining what the appropriate direction for movement is. It is then necessary to expand the bank to a coarse discretization that covers the entire parameter space, so that the true parameter becomes encompassed by the bank. This situation could occur in two scenarios: during initial acquisition and when the true parameter makes a discrete jump to a new location. The bank may be originally configured in a coarse discretization for the initial acquisition or expanded from a fine to a coarse discretization for a jump change. Hentz determined that there was substantial improvement in parameter estimation when the bank was initialized to a coarse discretization [26]. Once the acquisition cycle yields an estimate of the parameter position, the bank can be contracted to a fine discretization, and centered on the parameter estimate.

Expanding the bank requires the use of the *residual monitoring technique* since the other techniques rely on $\hat{\mathbf{a}}$, which is confined within the finely discretized bank by the computation of Equation (2.29), with summation only over the currently active parameter points. The decision to expand the bank is based upon the likelihood quotient, similar to the residual monitoring move logic discussed in Section 2.4.1.1. An expansion threshold is set (higher than the move threshold) such that if this value is exceeded by *all* the filters, then this indicates the parameter position has made a much larger jump. None of the current filters will have a viable hypothesized parameter value, hence an expansion to a coarse level of discretization is appropriate. As in the move method, the same problems exist for the expansion logic. Specifically, the method is dependent on only the single most recent measurement value, and thus is susceptible to a single large sample of measurement noise. This may cause the bank to expand unnecessarily. The expansion threshold is determined through trial and error and is very important to the proper

operation of the MMAE. Hentz found that if the expansion threshold was set too high, the response time was too slow. Likewise, if the threshold was set too low, the bank would expand unnecessarily [9:69].

2.4.3 Contracting the Bank

Contraction of the discretized bank will be appropriate once an accurate estimate of the uncertain parameter is obtained following either initial acquisition or previous expansion due to a jump parameter change. The bank is contracted to surround the true parameter with a fine configuration based on a technique called the *Parameter Estimation Error Covariance Monitoring* method. When the variance of the scalar parameter estimate (as determined in Equation (2.30)) falls below a set threshold, the bank can be contracted down to a fine configuration about the parameter estimate. Depending upon the size of the parameter space, this contraction can occur in one large step or multiple steps. As with the move and expansion logic, determination of the correct contraction threshold is critical to the MMAE performance. It is desirable to have the contraction occur quickly to enhance the parameter estimation, but Hentz found that if the threshold was set too high, contraction would occur before a good parameter estimate was obtained. Likewise, setting the threshold too low delayed proper contraction [9:69].

If the parameter space is greater than one dimensional, a problem occurs with the above procedure. In the problem at hand, a three-dimensional parameter space is used. A scalar threshold value can only be compared to some scalar measure of size associated with a three-by-three covariance matrix, $\mathbf{P}_{\hat{\mathbf{a}}}$. Hentz compared the largest diagonal element in the matrix with the threshold, whereas Filios had better results by requiring both diagonal elements to be below the threshold [2,9]. Unfortunately, this method has tradeoffs since the appropriate threshold for each diagonal element might be different. Better performance may be gained by allowing separate decisions in each of the parameter

space directions, which may lead to rectangular bank shapes as well as square. This is the method that will be implemented here.

Karnick [10] proposed an alternative method. His technique monitored the probability of an entire side of the bank as defined in the following equation:

$$p_{side}(t_i) = \frac{\sum_{side} f_{\mathbf{z}(t_i)|\mathbf{a},\mathbf{Z}(t_{i-1})}(\mathbf{z}_i | \mathbf{a}_j, \mathbf{Z}_{i-1})}{\sum_{4sides} f_{\mathbf{z}(t_i)|\mathbf{a},\mathbf{Z}(t_{i-1})}(\mathbf{z}_i | \mathbf{a}_j, \mathbf{Z}_{i-1})} \tag{2.33}$$

If this probability fell below the threshold, that particular side was contracted towards the parameter estimate. Likewise, if the probability of a side should rise above another threshold, then the other sides were contracted towards the parameter estimate [10:29]. Another variation monitored all four sides such that if the sum of these probabilities fell below a threshold, the bank was contracted. This method corresponds to the idea that the larger the probability, the closer the side is to the true parameter value.

2.4.4 Initialization of New Elemental Filters

Whenever the bank moves, expands, or contracts, new elemental filters will be activated. These newly activated filters must undergo an initialization process. First, $\mathbf{\Phi}$, $\mathbf{B}_d$, $\mathbf{G}_d$, $\mathbf{H}$, $\mathbf{A}$, $\mathbf{D}$, $\mathbf{K}$, $\mathbf{P}(t_i^-)$ and $\mathbf{P}(t_i^+)$ for the filter and $\overline{\mathbf{G}}_c^*$ for the controller (to be discussed in the next section) must be modified to correspond to the new parameter value. The matrix, $\mathbf{D}$, is a direct feedthrough matrix that results from the reduced order model development and will be discussed in Chapter 3. These matrices are updated from information stored in memory that corresponds to the new parameter value for the filters that now comprise the bank. The new filters also must have an appropriate state estimate, $\hat{\mathbf{x}}_j(t_i)$, and probability weight, $p_j(t_i)$, which must be initialized. The state estimate, $\hat{\mathbf{x}}_j(t_i)$, is set equal to the current adaptive state estimate, $\hat{\mathbf{x}}(t_i)$. There are several methods proposed to initialize $p_j(t_i)$, but the most widely used method involves redistributing the values last obtained from the deactivated filters *equally* among the newly

activated filters. Another method reinitializes all the current filters (changed and unchanged) to an equivalent probability weighting (all p_j values = $1/K$). This proved to converge to a parameter estimate too slowly [9,26]. However, if the bank expands or contracts, setting all values equal may be the most appropriate despite the slowness, since the old values may no longer be valid. A note to remember is that the sum of the new probability weights in all cases must add to one. Other, more complicated methods of distributing the probabilities have been examined [9:29], however, they were not found to be any more effective and thus will not be pursued for this research.

2.5 Stochastic Controller Development

Linear (system model), quadratic (cost function), Gaussian (noise model) techniques will be the main controller development tools for this research. It was stated in Chapter 1 that this thesis will use an LQG controller. For this type of controller, the quadratic cost function to be minimized can be given as:

$$J = E\left\{\begin{array}{l}\sum_{i=0}^{N}\frac{1}{2}\left[\mathbf{x}^T(t_i)\mathbf{X}(t_i)\mathbf{x}(t_i)+\mathbf{u}^T(t_i)\mathbf{U}(t_i)\mathbf{u}(t_i)+2\mathbf{x}^T(t_i)\mathbf{S}(t_i)\mathbf{u}(t_i)\right] \\ +\frac{1}{2}\mathbf{x}^T(t_{N+1})\mathbf{X}_f\mathbf{x}(t_{N+1})\end{array}\right\} \quad (2.34)$$

where the resulting output of the controller is the optimal control function, $\mathbf{u}^*$ [22:73]. This equation can be restated as:

$$J = E\left\{\sum_{i=0}^{N}\frac{1}{2}\left(\begin{bmatrix}\mathbf{x}(t_i)\\ \mathbf{u}(t_i)\end{bmatrix}^T\begin{bmatrix}\mathbf{X}(t_i) & \mathbf{S}(t_i)\\ \mathbf{S}^{\mathrm{T}}(t_i) & \mathbf{U}(t_i)\end{bmatrix}\begin{bmatrix}\mathbf{x}(t_i)\\ \mathbf{u}(t_i)\end{bmatrix}\right)+\frac{1}{2}\mathbf{x}^T(t_{N+1})\mathbf{X}_f\mathbf{x}(t_{N+1})\right\} \quad (2.35)$$

where:

- J = scalar cost to be minimized
- $\mathbf{x}(t_i)$ = n-dimensional state vector
- $\mathbf{X}(t_i)$ = n x n-dimensional state weighting matrix
- $\mathbf{X}_f$ = n x n-dimensional final state weighting matrix
- $\mathbf{u}(t_i)$ = r-dimensional control input vector
- $\mathbf{U}(t_i)$ = r x r-dimensional control weighting matrix
- $\mathbf{S}(t_i)$ = n x r-dimensional cross-weighting matrix
- t_N = last time a control is applied (held constant to next sample period)
- t_{N+1} = final time

$\mathbf{X}(t_i)$ and $\mathbf{X}_f$ are weighting matrices that are real, symmetric and positive semi-definite, which allows for a zero cost to be placed on states that have negligible impact on desired performance. $\mathbf{U}(t_i)$ is real, symmetric and positive definite, thus assuring that there are no zero eigenvalues, in order to preclude allowing infinite power to be commanded through any of the actuators. The cross weighting matrix $\mathbf{S}(t_i)$ is chosen so that the augmented matrix in Equation (2.35) is positive semi-definite, which ensures that a cost-minimizing controller solution exists.

For the purposes of this discussion, the weighting matrices are assumed to be diagonal. The values in the cost weighting matrices, $\mathbf{X}(t_i)$ and $\mathbf{X}_f$, reflect the relative importance of maintaining the state estimates near zero. Larger terms indicate higher emphasis on minimizing the corresponding state. Similarly, the diagonal terms of the control weighting matrix, $\mathbf{U}(t_i)$, reflect the level of control energy or power that is to be used. Larger terms indicate smaller levels of control power are to be commanded through the corresponding actuators.

For the quadratic cost defined in Equation (2.35), the optimal discrete linear feedback control law is [22:16]:

$$\mathbf{u}^*(t_i) = -\mathbf{G}_c^*(t_i)\hat{\mathbf{x}}(t_i^+) \tag{2.36}$$

where $\mathbf{G}_c^*(t_i)$ is determined by solving a backward Riccati difference equation and $\hat{\mathbf{x}}(t_i^+)$ is the state estimate from the Kalman filter.

A steady-state constant-gain control law can be obtained if the initial transients in the Kalman filter and the final transients in the controller gain are assumed to have minimal impact on the system performance over the total time of interest. For time-invariant systems and constant weighting matrices (i.e., **X**, **U**, and **S** are not functions of time in Equation (2.35)), steady-state, constant-gain controllers can be generated by ignoring the final transient, notationally by letting the upper limit, *N*, go to infinity and dropping the last term in Equation (2.35). In this case, the optimal discrete linear feedback control law is [22:243]:

$$\mathbf{u}^*(t_i) = -\overline{\mathbf{G}}_c^*\hat{\mathbf{x}}(t_i^+) \tag{2.37}$$

where $\hat{\mathbf{x}}(t_i^+)$ is now the state estimate from the constant-gain, steady-state Kalman filter.

The constant-gain steady state control law is as given by Equation (2.37), with [22:242]:

$$\overline{\mathbf{G}}_c^* = \left[\mathbf{U} + \mathbf{B}_d^{\mathrm{T}}\overline{\mathbf{K}}_c\mathbf{B}_d\right]^{-1}\left[\mathbf{B}_d^{\mathrm{T}}\overline{\mathbf{K}}_c\boldsymbol{\Phi} + \mathbf{S}^{\mathrm{T}}\right] \tag{2.38}$$

where $\overline{\mathbf{K}}_c^*$ is determined by solving the steady-state Riccati equation:

$$\overline{\mathbf{K}}_c = \mathbf{X} + \boldsymbol{\Phi}^{\mathrm{T}}\overline{\mathbf{K}}_c\boldsymbol{\Phi} - \left[\mathbf{B}_d^{\mathrm{T}}\overline{\mathbf{K}}_c\boldsymbol{\Phi} + \mathbf{S}^{\mathrm{T}}\right]^{\mathrm{T}}\left[\mathbf{U} + \mathbf{B}_d^{\mathrm{T}}\overline{\mathbf{K}}_c\mathbf{B}_d\right]^{-1}\left[\mathbf{B}_d^{\mathrm{T}}\overline{\mathbf{K}}_c\boldsymbol{\Phi} + \mathbf{S}^{\mathrm{T}}\right] \tag{2.39}$$

These assumptions reduce the quadratic cost function to:

$$J = E\left\{\sum_{i=0}^{\infty}\tfrac{1}{2}\left[\mathbf{x}^{\mathrm{T}}(t_i)\mathbf{X}\mathbf{x}(t_i) + \mathbf{u}^{\mathrm{T}}(t_i)\mathbf{U}\mathbf{u}(t_i) + 2\mathbf{x}^{\mathrm{T}}(t_i)\mathbf{S}\mathbf{u}(t_i)\right]\right\} \tag{2.40}$$

Previous research results indicated that the cross weighting matrix, **S**, had a negligible magnitude, and it was neglected with no appreciable performance impact [13]. This is the form of control law to be used in this thesis.

2.6 MMAC

The LQG controller MMAC design is similar to the MMAE development, and was depicted in Figure 1-2. One LQG controller as described in the previous section will be developed for each Kalman filter in the bank. The controller gain , $-\overline{\mathbf{G}}_c^*$, is cascaded with the Kalman filter, however, there are many methods in which the final control is implemented. As introduced in Chapter 1, there are six methods for combining the MMAE and LQG controller. These are: (1) MMAC Control, (2) "Modified" MMAC Control, (3) MAP versus Bayesian MMAC Control, (4) Single Fixed-Gain Control, (5) Single Changeable-Gain Control, and (6) "Modified" Single Changeable-Gain Control.

2.6.1 MMAC Control

The pure MMAC control method is shown in Figure 1-2. The final optimal control output is determined in much the same manner as is the Bayesian form of the optimal state estimate in the MMAE. Each controller output is appropriately weighted-based on the filter conditional probability, $p_k(t_i)$, and summed with the others to create a weighted-average final controller output. Note that each controller gain is based on the assumed parameter value for that filter model.

2.6.2 "Modified" MMAC Control

The modified MMAC control method ignores inputs from filter/controllers assumed to have relatively poor state estimates by monitoring the conditional probabilities, $p_k(t_i)$. After each of the K control vectors, $\mathbf{u}_k$, (see Figure 1-2) is formed, the associated conditional probabilities, $p_k(t_i)$, are compared to a set threshold. If the probability of any filter falls below this threshold, its corresponding control output is not included in the final summed controller gain. However, the probabilities for the other controllers must be rescaled to sum to one. Care must be taken in setting this threshold; it must be set higher than the artificial lower bound which was added to the conditional probabilities to keep

them from ever going to zero. By applying another threshold on the conditional probabilities, higher than the previously mentioned lower bound, filters at the lower bound or in the gap between the two bounds will be effectively eliminated from the MMAC control calculation. The intent is to remove contributions to the MMAC output that are derived from elemental LQG controllers based on wrong hypothesized parameter values, since these would likely be inappropriate or even destabilizing control contributions.

2.6.3 MAP verses Bayesian MMAC Control

This method, which corresponds to the MAP estimation described in Section 2.3.2, states that the elemental LQG controller with the largest computed conditional probability will be declared the controller to be used at the current time. Again, this method doesn't allow for "blending" of the controllers.

2.6.4 Single Fixed-Gain Control

This method determines the control gain based on the nominal parameter values. This is assumed to be acceptable since the full-state feedback controller is inherently robust [9:40]. Figure 2-1 illustrates this design. In this figure, ZOH refers to zero-order-hold, where the control input is held constant over the entire sample period until a new control input is determined. Likewise, T refers to the discrete time sampler period. The control law is given by:

$$\mathbf{u}^*(t_i) = -\overline{\mathbf{G}}_c^*[\mathbf{a}_{nom}]\hat{\mathbf{x}}(t_i^+) \tag{2.41}$$

where the state estimate is the only input and the gain is pre-computed. Determination of $\mathbf{a}_{nom}$ is not a trivial task in a real system, however, the controller parameters must be selected such that the controller provides regulation for any true system parameter value [9:40].

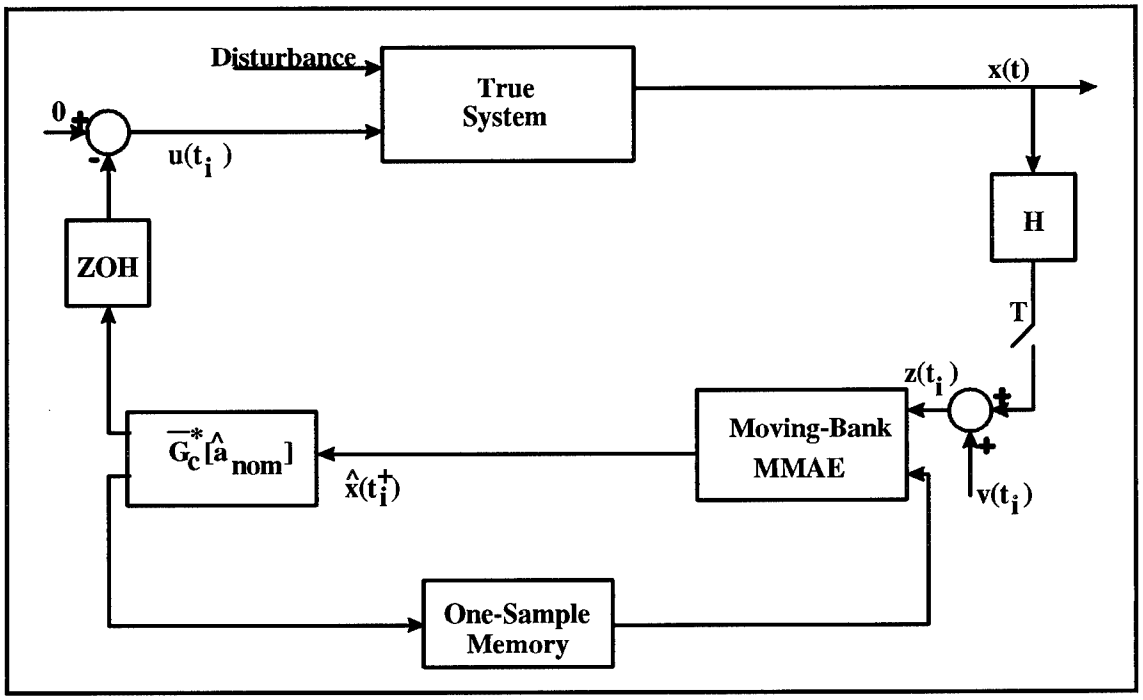


Figure 2-1. Single Fixed-Gain Controller

2.6.5 Single Changeable-Gain Control

This method is similar to the last in that it uses the state estimate as output from the MMAE, however, the controller gain is determined from the parameter estimate as output from the MMAE. This allows for a changing value for $\hat{\mathbf{a}}(t_i)$ and is illustrated in Figure 2-2. The governing control law is as follows:

$$\mathbf{u}^*(t_i) = -\overline{\mathbf{G}}_c^*[\hat{\mathbf{a}}(t_i)]\hat{\mathbf{x}}(t_i^+) \tag{2.42}$$

where the gain is determined as a function of the parameter position estimates. These gains can be precomputed for each of the K parameter values and stored in a look-up table which is accessed each time a new value of $\hat{\mathbf{a}}(t_i)$ is generated by the MMAE algorithm. Interpolation can be used to compute actual controller gains from such a look-up table [9:38].

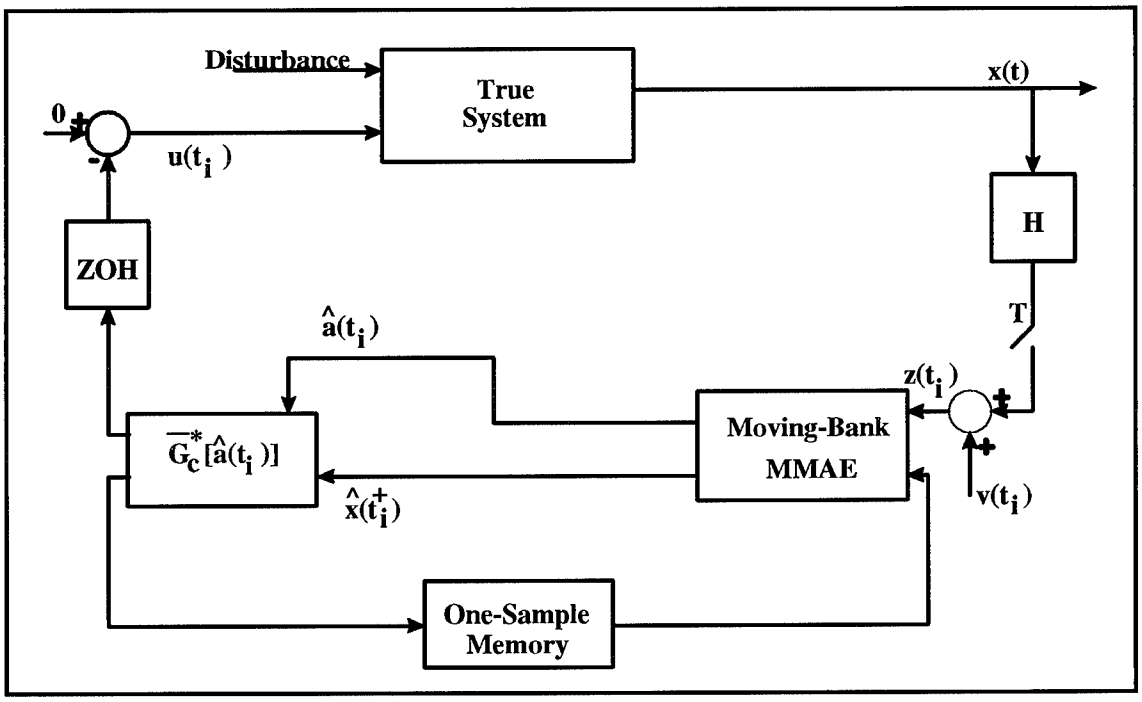


Figure 2-2. Single Changeable-Gain Controller

2.6.6 "Modified" Single Changeable-Gain Control

Similar to the previous method, the modified single changeable-gain technique also uses a state estimate and a parameter estimate to determine the control input. The parameter estimate continues to come from the moving bank MMAE, however, the state estimate comes from a separate Kalman filter which is not part of the moving bank of filters. $\hat{\mathbf{a}}(t_i)$ from the MMAE is provided to the single separate Kalman filter/controller in which the system model and controller gain is based on the estimated parameter position. This type of control can be useful if $\hat{\mathbf{a}}(t_i)$ lies between discretized values $\mathbf{a}_k$. Figure 2-3 illustrates this technique. For the previous method $\hat{\mathbf{x}}(t_i^+)$ in Figure 2-2 will be produced by weighted-average blending of a number of elemental filter outputs $\hat{\mathbf{x}}_k(t_i^+)$, none of which are based on the "correct" parameter value, whereas $\hat{\mathbf{x}}(t_i^+)$ of Figure 2-3 would be produced by a single Kalman filter explicitly based on that correct parameter value. Thus, discretization effects should cause less state estimation degradation in this type of controller than in the form depicted in Figure 2-2. This combination should reduce the

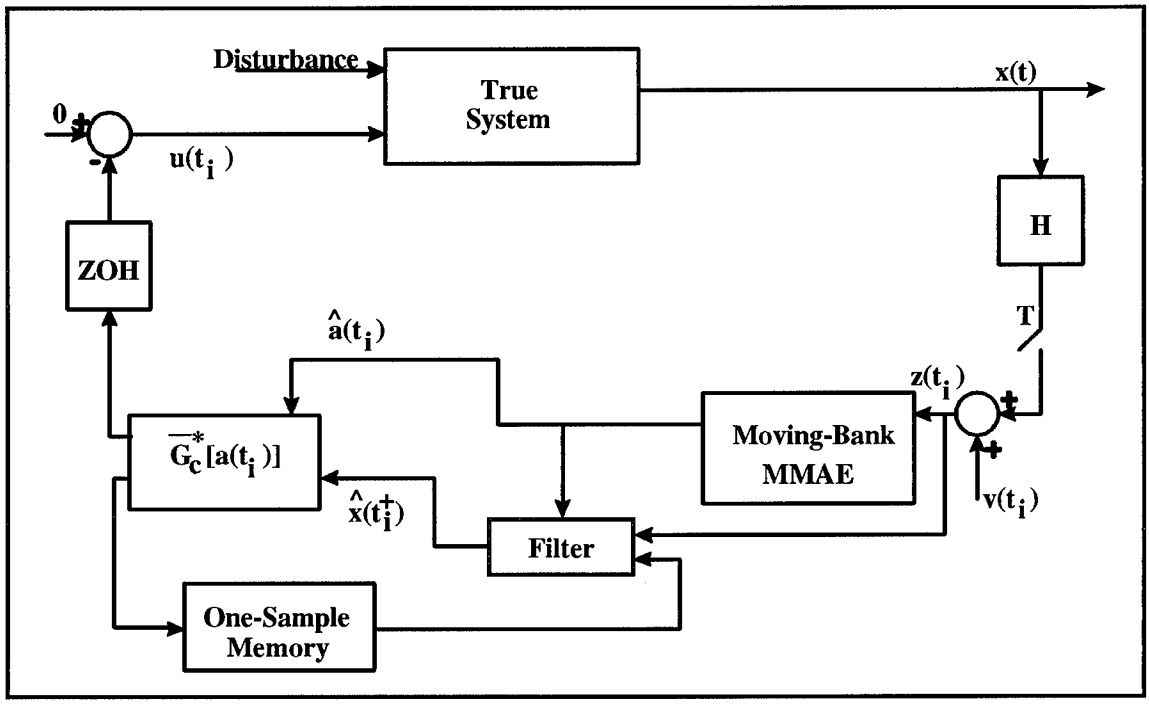


Figure 2-3. Modified Single Changeable-Gain Controller

probability that the control input is generated from filters and/or controllers that assume too small a value for the undamped natural frequency of the system [7:1-17]. Past research has shown that underestimating this value for the important bending modes will readily lead to system instabilities [33].

2.7 Mathematical Modeling Methods

A system model for a physical structure is usually determined from finite element analysis and testing. This results in the mass, damping and stiffness matrices which are used in the physical coordinate system. Unfortunately, this coordinate system results in highly coupled equations which are not readily comprehended. This section will provide a brief presentation of the physical coordinate form and then discuss the transformation into a more desirable modal form. State order reduction is typically desired for on-line filter/controller implementations due to the excessive computational burden imposed by high dimensioned systems. Consequently, two techniques for accomplishing state order

reduction will be presented. These are the modal reduction, and the modal cost reduction techniques. Additionally, the internally balanced reduction technique has been examined in the past, but the poor results obtained with this method do not warrant any further consideration [31].

2.7.1 Physical Coordinate Form

The dynamics of a structure's flexible body modes are described by the following standard second-order differential equation [10:40, 36:1769]:

$$\mathbf{M}\ddot{\mathbf{r}}(t)+\mathbf{C}\dot{\mathbf{r}}(t)+\mathbf{K}\mathbf{r}(t)=\mathbf{F}_1(\mathbf{u},t)+\mathbf{F}_2(t) \tag{2.43}$$

where:

- $\mathbf{r}(t)$ = n-dimensional vector representing the structure's physical position
- $\mathbf{M}$ = n-by-n constant mass matrix
- $\mathbf{C}$ = n-by-n constant damping matrix
- $\mathbf{K}$ = n-by-n constant stiffness matrix
- $\mathbf{F}_1(\mathbf{u},t)$ = r-dimensional deterministic control inputs
- $\mathbf{F}_2(t)$ = r-dimensional disturbances and unmodeled control inputs

$\mathbf{F}_2(t)$ can be modeled as white Gaussian noises. Under the assumption that these disturbances enter through linear, time-invariant matrices, then Equation (2.43) becomes [10:40, 36:1769]:

$$\mathbf{M}\ddot{\mathbf{r}}(t)+\mathbf{C}\dot{\mathbf{r}}(t)+\mathbf{K}\mathbf{r}(t)=-\mathbf{b}\mathbf{u}(t)-\mathbf{g}\mathbf{w}(t) \tag{2.44}$$

where:

- $\mathbf{u}(t)$ = r-dimensional vector actuator inputs

- $\mathbf{b}$ = n-by-r control input matrix identifying position and relationships between actuators and controlled variables
- $\mathbf{w}(t)$ = s-dimensional vector of white Gaussian noises representing the dynamics driving noise
- $\mathbf{g}$ = n-by-s noise input matrix identifying position and relationship between the dynamics driving noise and the controlled variables

Equation (2.44) can be transformed into the following state space form [10:40, 1:63]:

$$\dot{\mathbf{x}}(t) = \mathbf{F}\mathbf{x}(t) + \mathbf{B}\mathbf{u}(t) + \mathbf{G}\mathbf{w}(t) \tag{2.45}$$

which is the same form as the stochastic differential equation described by Equation (2.1) in Section 2.2. Thus, the state vector representation from Equation (2.44) of a general structure is given by:

$$\mathbf{x}(t) = \begin{bmatrix} \dot{\mathbf{r}}(t) \\ \mathbf{r}(t) \end{bmatrix}_{2nx1} \tag{2.46}$$

and the form of the constant system matrices is [7:3-20,10:41]:

$$\mathbf{F} = \begin{bmatrix} -\mathbf{M}^{-1}\mathbf{C}_{nxn} & -\mathbf{M}^{-1}\mathbf{K}_{nxn} \\ \mathbf{I}_{nxn} & \mathbf{0}_{nxn} \end{bmatrix}_{2nx2n} \tag{2.47}$$

$$\mathbf{B} = \begin{bmatrix} -\mathbf{M}^{-1}\mathbf{b}_{nxr} \\ \mathbf{0}_{nxr} \end{bmatrix}_{2nxr} \tag{2.48}$$

$$\mathbf{G} = \begin{bmatrix} -\mathbf{M}^{-1}\mathbf{g}_{nxs} \\ \mathbf{0}_{nxs} \end{bmatrix}_{2nxr} \tag{2.49}$$

For the state vector described in Equation (2.46), the description of discrete-time measurements of position and velocity is given by [7:3-20, 10:41]:

$$\mathbf{z}(t_i) = \left\{ \begin{bmatrix} \mathbf{H}_v & \mathbf{0} \\ \mathbf{0} & \mathbf{H}_p \end{bmatrix}_{mx2n} \mathbf{x}(t_i) \right\} + \mathbf{v}(t_i) \tag{2.50}$$

where:

- m = number of measurements
- $\mathbf{v}(t_i)$ = m-dimensional measurement uncertainty modeled as a discrete-time white Gaussian noise of covariance $\mathbf{R}(t_i)$
- $\mathbf{H}_p$ = (m/2)-by-n position measurement matrix in physical coordinates
- $\mathbf{H}_v$ = (m/2)-by-n velocity measurement matrix in physical coordinates

It is assumed that an equal number of position and velocity measurements are available, however, the actual measurement matrix may vary as a function of the measurements available. This will not affect the structure of the system matrices of Equations (2.47) through (2.49).

2.7.2 Modal Coordinate Form

The physical coordinate form described above has the disadvantage that the system equations are highly coupled, thus the important characteristics of the system are difficult to identify. Transforming the equations to modal form decouples the modes and makes the identification process simpler. In this research, the damping matrix, **C**, is assumed to be a linear combination of the mass and stiffness matrices [36:1769]:

$$\mathbf{C} = \alpha\mathbf{M} + \beta\mathbf{K} \tag{2.51}$$

However, the calculation of α and β is not necessary when transforming to the modal coordinate form, as will be seen. Given the new modal coordinate vector $\tilde{\mathbf{r}}$, the relationship between the modal and physical forms is described by [36:1769]:

$$\mathbf{r} = T\tilde{\mathbf{r}} \tag{2.52}$$

where $\boldsymbol{T}$ is a n-by-n transformation matrix determined from the system eigenvectors calculated from [36:1769]:

$$\omega^2 \mathbf{M}\boldsymbol{T} = \mathbf{K}\boldsymbol{T} \tag{2.53}$$

where the values for ω that satisfy this equation are referred to as the natural or modal frequencies. Since the damping matrix $\mathbf{C}$ does not appear in this equation, the previous statement that the parameters in Equation (2.51) are not required is shown to be true.

Now, using the transformation Equation (2.52) to operate on the original system Equation (2.45), the resulting state space equation is given by [17:5]:

$$\dot{\tilde{\mathbf{x}}}(t) = \tilde{\mathbf{F}}\tilde{\mathbf{x}}(t) + \tilde{\mathbf{B}}\mathbf{u}(t) + \tilde{\mathbf{G}}\mathbf{w}(t) \tag{2.54}$$

where the transformed state vector from Equation (2.46) is now defined as [17:5]:

$$\tilde{\mathbf{x}}(t) = \begin{bmatrix} \dot{\tilde{\mathbf{r}}}(t) \\ \tilde{\mathbf{r}}(t) \end{bmatrix}_{2nx1} \tag{2.55}$$

and the transformed matrices from Equation (2.45) as applied to Equation (2.44) are defined as [7, 17:5]:

$$\tilde{\mathbf{F}} = \begin{bmatrix} -\boldsymbol{T}^{-1}\mathbf{M}^{-1}\mathbf{C}\boldsymbol{T} & -\boldsymbol{T}^{-1}\mathbf{M}^{-1}\mathbf{K}\boldsymbol{T} \\ \mathbf{I} & \mathbf{0} \end{bmatrix}_{2nx2n} \tag{2.56}$$

$$\tilde{\mathbf{B}} = \begin{bmatrix} -\boldsymbol{T}^{-1}\mathbf{M}^{-1}\mathbf{b} \\ \mathbf{0} \end{bmatrix}_{2nxr} = \tilde{\mathbf{G}} \tag{2.57}$$

$$\tilde{\mathbf{G}} = \begin{bmatrix} -\boldsymbol{T}^{-1}\mathbf{M}^{-1}\mathbf{g} \\ \mathbf{0} \end{bmatrix}_{2nxr} \tag{2.58}$$

Since the transformation matrix is developed from the system eigenvectors which are orthogonal, the modal form results in independent equations. This fact plus the following relationships:

$$-\boldsymbol{T}^{-1}\mathbf{M}^{-1}\mathbf{C}\boldsymbol{T} = \left[-2\zeta_i\omega_i\right] \tag{2.59}$$

$$-\boldsymbol{T}^{-1}\mathbf{M}^{-1}\mathbf{K}\boldsymbol{T} = \left[-\omega_i^2\right] \tag{2.60}$$

allows the dynamics matrix to be written as follows:

$$\tilde{\mathbf{F}} = \begin{bmatrix} \left[-2\zeta_i\omega_i\right] & \left[-\omega_i{}^2\right] \\ \mathbf{I} & \mathbf{0} \end{bmatrix}_{2nx2n} \tag{2.61}$$

where the upper partitions are now block diagonal in terms of the undamped natural frequency and the damping ratio of the *i*-th mode. The transformed measurement equation from Equation (2.50) may be written as [7]:

$$\mathbf{z}(t_i) = \left\{ \begin{bmatrix} \mathbf{H}_v\boldsymbol{T} & \mathbf{0} \\ \mathbf{0} & \mathbf{H}_p\boldsymbol{T} \end{bmatrix}_{mx2n} \tilde{\mathbf{x}}(t_i) \right\} + \mathbf{v}(t_i) \tag{2.62}$$

Thus, the modal form, unlike the physical form, provides a decoupled set of system equations which allow ready access to the individual modes of the system (in terms of the natural frequency and damping ratio). The modal form provides more insight to the physical structure in applicable large order systems. However, the method cannot be used if the plant matrices are time varying [7].

2.7.3 Modal Reduction Technique

The modal reduction technique is one way to reduce the number of states in the system model such that on-line applicability can be attained. In modal reduction, the higher frequency modes are eliminated from the system model. The elimination of high frequency modes comes from the assumption that, at the higher frequencies, the structure reaches steady state in a negligibly small amount of time. Also, due to the physical shape of the SPICE structure, the lower frequency modes should dominate the response of the system. This section will discuss the modal reduction procedure.

From Equation (2.45), the continuous state-space system model can be partitioned as follows [10:52,12:124]:

$$\dot{\mathbf{x}}(t)=\begin{bmatrix}\dot{\mathbf{x}}_1(t)\\ \dot{\mathbf{x}}_2(t)\end{bmatrix}=\begin{bmatrix}\mathbf{F}_{11} & \mathbf{F}_{12}\\ \mathbf{F}_{21} & \mathbf{F}_{22}\end{bmatrix}\begin{bmatrix}\mathbf{x}_1(t)\\ \mathbf{x}_2(t)\end{bmatrix}+\begin{bmatrix}\mathbf{B}_1\\ \mathbf{B}_2\end{bmatrix}\mathbf{u}(t)+\begin{bmatrix}\mathbf{G}_1\\ \mathbf{G}_2\end{bmatrix}\mathbf{w}(t) \tag{2.63}$$

where the system is driven by deterministic controls, $\mathbf{u}(t)$, and zero-mean, white Gaussian noise $\mathbf{w}(t)$ of strength $\mathbf{Q}(t)$. The upper partition, $\mathbf{x}_1(t)$, corresponds to the low frequency modes to be maintained, and the lower partition, $\mathbf{x}_2(t)$, corresponds to the high frequency modes to be removed.

Assuming instantaneous steady state ($\dot{\mathbf{x}}_2(t)=\mathbf{0}$), the $\mathbf{x}_2(t)$ modes can be eliminated with negligible impact to the overall performance of the system. The lower partition differential equation is set to zero as follows [10:52, 12:124]:

$$\dot{\mathbf{x}}_2(t)=\mathbf{F}_{21}\mathbf{x}_1(t)+\mathbf{F}_{22}\mathbf{x}_2(t)+\mathbf{B}_2\mathbf{u}(t)+\mathbf{G}_2\mathbf{w}(t)=\mathbf{0} \tag{2.64}$$

Since $\mathbf{F}_{21}$ and $\mathbf{F}_{22}$ are square matrices and $\mathbf{F}_{22}^{-1}$ is assumed to exist, $\mathbf{x}_2(t)$ can now be written in terms of $\mathbf{x}_1(t)$ and system inputs [10:52, 12:123]:

$$\mathbf{x}_2(t)=-\mathbf{F}_{22}^{-1}\left[\mathbf{F}_{21}\mathbf{x}_1(t)+\mathbf{B}_2\mathbf{u}(t)+\mathbf{G}_2\mathbf{w}(t)\right] \tag{2.65}$$

Substituting Equations (2.64) and (2.65) into Equation (2.63) results in [10:52, 12:124]:

$$\begin{aligned}\dot{\mathbf{x}}_1(t)=&\left[\mathbf{F}_{11}-\mathbf{F}_{12}\mathbf{F}_{22}^{-1}\mathbf{F}_{21}\right]\mathbf{x}_1(t)+\left[\mathbf{B}_1-\mathbf{F}_{12}\mathbf{F}_{22}^{-1}\mathbf{B}_2\right]\mathbf{u}(t)\\ &+\left[\mathbf{G}_1-\mathbf{F}_{12}\mathbf{F}_{22}^{-1}\mathbf{G}_2\right]\mathbf{w}(t)\end{aligned} \tag{2.66}$$

Since this research will use a digital implementation, the discrete-time version of this procedure is derived in a similar fashion [7:3-24]. The equivalent discrete-time model of Equation (2.63) is as follows:

$$\mathbf{x}(t_{i+1})=\begin{bmatrix}\mathbf{x}_1(t_{i+1})\\ \mathbf{x}_2(t_{i+1})\end{bmatrix}=\begin{bmatrix}\mathbf{\Phi}_{11} & \mathbf{\Phi}_{12}\\ \mathbf{\Phi}_{21} & \mathbf{\Phi}_{22}\end{bmatrix}\begin{bmatrix}\mathbf{x}_1(t_i)\\ \mathbf{x}_2(t_i)\end{bmatrix}+\begin{bmatrix}\mathbf{B}_{d1}\\ \mathbf{B}_{d2}\end{bmatrix}\mathbf{u}(t_i)+\begin{bmatrix}\mathbf{G}_{d1}\\ \mathbf{G}_{d2}\end{bmatrix}\mathbf{w}_d(t_i) \tag{2.67}$$

With the same steady state assumption as applied to the discrete time case, $(\mathbf{x}_2(t_{i+1}) = \mathbf{x}_2(t_i))$, the resulting representation of the higher order modes is given as follows:

$$\mathbf{\Phi}_{21}\mathbf{x}_1(t_i) + \left[\mathbf{\Phi}_{22} - \mathbf{I}\right]\mathbf{x}_2(t_i) + \mathbf{B}_{d2}\mathbf{u}(t_i) + \mathbf{G}_{d2}\mathbf{w}_d(t_i) = \mathbf{0} \tag{2.68}$$

$$\mathbf{x}_2(t_i) = -\left[\mathbf{\Phi}_{22} - \mathbf{I}\right]^{-1}\left[\mathbf{\Phi}_{21}\mathbf{x}_1(t_i) + \mathbf{B}_{d2}\mathbf{u}(t_i) + \mathbf{G}_{d2}\mathbf{w}_d(t_i)\right] \tag{2.69}$$

The continuous state-space equation can be discretized easily by assuming a first order approximation as follows:

$$\mathbf{\Phi}_{22} \cong \mathbf{I} + \mathbf{F}_{22}\Delta t \tag{2.70}$$

$$\mathbf{\Phi}_{21} \cong \mathbf{F}_{21}\Delta t \tag{2.71}$$

$$\mathbf{B}_{d2} \cong B_2\Delta t \tag{2.72}$$

$$\mathbf{G}_{d2} \cong \mathbf{G}_2\Delta t \tag{2.73}$$

$$\mathbf{Q}_d \cong \mathbf{Q} / \Delta t \tag{2.74}$$

This last equation comes from the first order approximation, $\mathbf{G}_{d2}\mathbf{Q}_d\mathbf{G}_{d2}^{\mathrm{T}} \cong \mathbf{G}_2\mathbf{Q}\mathbf{G}_2^{\mathrm{T}}\Delta t$ and $\mathbf{G}_{d2} = \mathbf{G}_2\Delta t$.

Substituting the Equations (2.70) through (2.74) into Equation (2.69) results in:

$$\mathbf{x}_2(t_i) = -\left[\mathbf{F}_{22}\Delta t\right]^{-1}\left[\mathbf{F}_{21}\Delta t\mathbf{x}_1(t_i) + \mathbf{B}_2\Delta t\mathbf{u}(t_i) + \mathbf{G}_2\Delta t\mathbf{w}_d(t_i)\right] \tag{2.75}$$

where $\mathbf{w}_d(t_i)$ has covariance $\mathbf{Q}_d = \mathbf{Q} / \Delta t$. Simplifying the previous equation results in:

$$\mathbf{x}_2(t_i) = -\mathbf{F}_{22}^{-1}\left[\mathbf{F}_{21}\mathbf{x}_1(t_i) + \mathbf{B}_2\mathbf{u}(t_i) + \mathbf{G}_2\mathbf{w}_d(t_i)\right] \tag{2.76}$$

This equation is similar to the previous continuous time version, Equation (2.65), except $\mathbf{w}$ is replaced with $\mathbf{w}_d$, a discrete-time white Gaussian noise. Then, substituting Equation (2.76) into the discrete time measurement equation:

$$\mathbf{z}(t_i) = \begin{bmatrix} \mathbf{H}_1 & \mathbf{H}_2 \end{bmatrix} \begin{bmatrix} \mathbf{x}_1(t_1) \\ \mathbf{x}_2(t_2) \end{bmatrix} + \mathbf{v}(t_i) \tag{2.77}$$

and expanding, yields:

$$\mathbf{z}(t_i) = \left[\mathbf{H}_1 - \mathbf{H}_2\mathbf{F}_{22}^{-1}\mathbf{F}_{21}\right]\mathbf{x}_1(t_i) - \mathbf{H}_2\mathbf{F}_{22}^{-1}\left[\mathbf{B}_2\mathbf{u}(t_i) + \mathbf{G}_2\mathbf{w}_d(t_i)\right] + \mathbf{v}(t_i) \tag{2.78}$$

which results in a direct feedthrough term (the second term) created by the order reduction [10:52, 12:124].

The determination of the break point between the desired low frequency modes and the unwanted high frequency modes is determined by the computational loading restrictions and from physical insight. This is accomplished by examining a list of the modal frequencies and determining the natural breaks in the groupings. The preserved low frequency modes form the basis of the reduced-order model, which still retains some information from the eliminated higher frequency modes .

Once a break point for the order reduction is determined, the system dynamics matrix, presented in Equation (2.61) is partitioned as follows to illustrate the modal form further [7:3-26, 10:53, 12:124]:

$$\tilde{\mathbf{F}} = \left[\begin{array}{cc|cc} \left[-2\zeta_1\omega_1\right] & \left[-\omega_1^2\right] & \mathbf{0} & \mathbf{0} \\ \mathbf{I} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \hline \mathbf{0} & \mathbf{0} & \left[-2\zeta_2\omega_2\right] & \left[-\omega_2^2\right] \\ \mathbf{0} & \mathbf{0} & \mathbf{I} & \mathbf{0} \end{array}\right] \tag{2.79}$$

The low frequency modes to be maintained are represented in the upper left quadrant, and the higher frequency modes to be eliminated are represented in the lower right quadrant. These two quadrants correspond to the $\mathbf{F}_{11}$ and $\mathbf{F}_{22}$ partitions in Equation (2.63). The off-diagonal blocks, $\mathbf{F}_{12}$ and $\mathbf{F}_{21}$, are zero. Substituting this information into Equations (2.66) and (2.78) yields [10:53, 12:124]:

$$\dot{\tilde{\mathbf{x}}}_1(t) = \tilde{\mathbf{F}}_{11}\tilde{\mathbf{x}}_1(t) + \tilde{\mathbf{B}}_1\mathbf{u}(t) + \tilde{\mathbf{G}}_1\mathbf{w}_r(t) = \tilde{\mathbf{F}}_r\tilde{\mathbf{x}}_1(t) + \tilde{\mathbf{B}}_r\mathbf{u}(t) + \tilde{\mathbf{G}}_r\mathbf{w}_r(t) \quad (2.80)$$

$$\begin{aligned}\mathbf{z}(t_i) &= \tilde{\mathbf{H}}_1\tilde{\mathbf{x}}_1(t_i) - \tilde{\mathbf{H}}_2\tilde{\mathbf{F}}_{22}^{-1}\left[\tilde{\mathbf{B}}_2\mathbf{u}(t_i) + \tilde{\mathbf{G}}_2\mathbf{w}_d(t_i)\right] + \mathbf{v}_r(t_i) \\ &= \tilde{\mathbf{H}}_r\tilde{\mathbf{x}}_1(t_i) - \tilde{\mathbf{D}}_u\mathbf{u}(t_i) + \tilde{\mathbf{D}}_w\mathbf{w}_d(t_i) + \mathbf{v}_r(t_i) \quad (2.81)\end{aligned}$$

where the subscript *r* denotes "reduced-order." As can be seen through this development, the direct feedthrough terms, $\tilde{\mathbf{D}}_u$ and $\tilde{\mathbf{D}}_w$, allow direct measurement of the effects of control inputs, $\mathbf{u}(t_i)$, and system dynamics driving noise, $\mathbf{w}_d(t_i)$, by the reduced order models.

2.7.4 Component Cost Modal Reduction Technique

Skelton and Yousuff [34] developed a method for order reduction called the Component Cost Analysis (CCA) technique. This method suggests that the performance of a dynamic system subject to white noise disturbances can be evaluated in terms of a *performance metric*, $\mathcal{V}$. This scalar value could represent the total system energy. The method determines what fraction of the overall system performance metric $\mathcal{V}$ is due to each component of the system. By identifying the individual cost associated with each state, it is possible to consider order reduction based on eliminating those states that have the least impact to the overall cost. A complete description of this technique can be found in Skelton and Yousuff [34]. This method of order reduction was used by Schiller [31] however, it will not be used in this research.

2.8 Summary

This chapter presented the fundamental concepts that are the building blocks of this research. The underlying assumptions for the entire development are: (1) the system is modeled by linear, time-invariant models driven by white, Gaussian noises, (2) LQG controller weighting matrices for state estimates and control inputs are assumed constant,

and cross weighting matrices are deemed negligible and are eliminated, and (3) steady-state Kalman filter and LQG controller gains are employed.

First, the basics of the Kalman filter were presented, followed by the development of the Bayesian MMAE, including the concept of a moving bank. The moving-bank MMAE discussion included methods for moving, expanding and contracting the bank. Next, the stochastic LQG controller development was provided with the subsequent application to the MMAC. The various methods of implementing the MMAE and the MMAC were described. The two coordinate forms used in this research were discussed, as was the method of transforming between the two systems. Finally, two methods for state order reduction were presented. Chapter 3 will proceed with a description of the SPICE-4 system structure model to be used for this research.

III. System Development

3.1 Introduction

This chapter presents a more thorough description of the SPICE-4 system model than was given in Chapter 1. This model was originally developed by Schiller [31]. First, a brief physical description of the structure is provided, followed by a description of the mathematical models used to simulate the structure and its inputs. Finally, the truth model and subsequent reduced-order design model are discussed.

3.2 SPICE Structure

The structural model used for this research is version 4 of the SPace Integrated Controls Experiment (SPICE). The actual structure is located at Phillips Laboratory, Kirtland AFB, New Mexico. SPICE has currently been revised up to version 7, but the system models were not available at the time this research was initiated. The SPICE structure must be able to be rotated or slewed perpendicular to the LOS axis by an active rigid body control system. Once slewed, the flexible body modes must be quelled to within a predetermined specification along the LOS axis to maintain the capability of precision pointing along the line of sight (LOS) axis (See Figure 3-1). This thesis will focus on quelling the flexible body vibrations and ignores rigid body motion effects.

3.2.1 Physical Structure Description

The SPICE structure is divided into three major structural sections as depicted in Figure 3-1. The hexagonal base, or Bulkhead, forms the support for the entire structure and is 6.19 meters in diameter. The Primary Mirror (PM) Assembly is mounted on top of the bulkhead. Three legs (tripod) connect the bulkhead to the Secondary Mirror (SM) Assembly, which is 1.32 meters in diameter. The overall height of the structure is 8.14

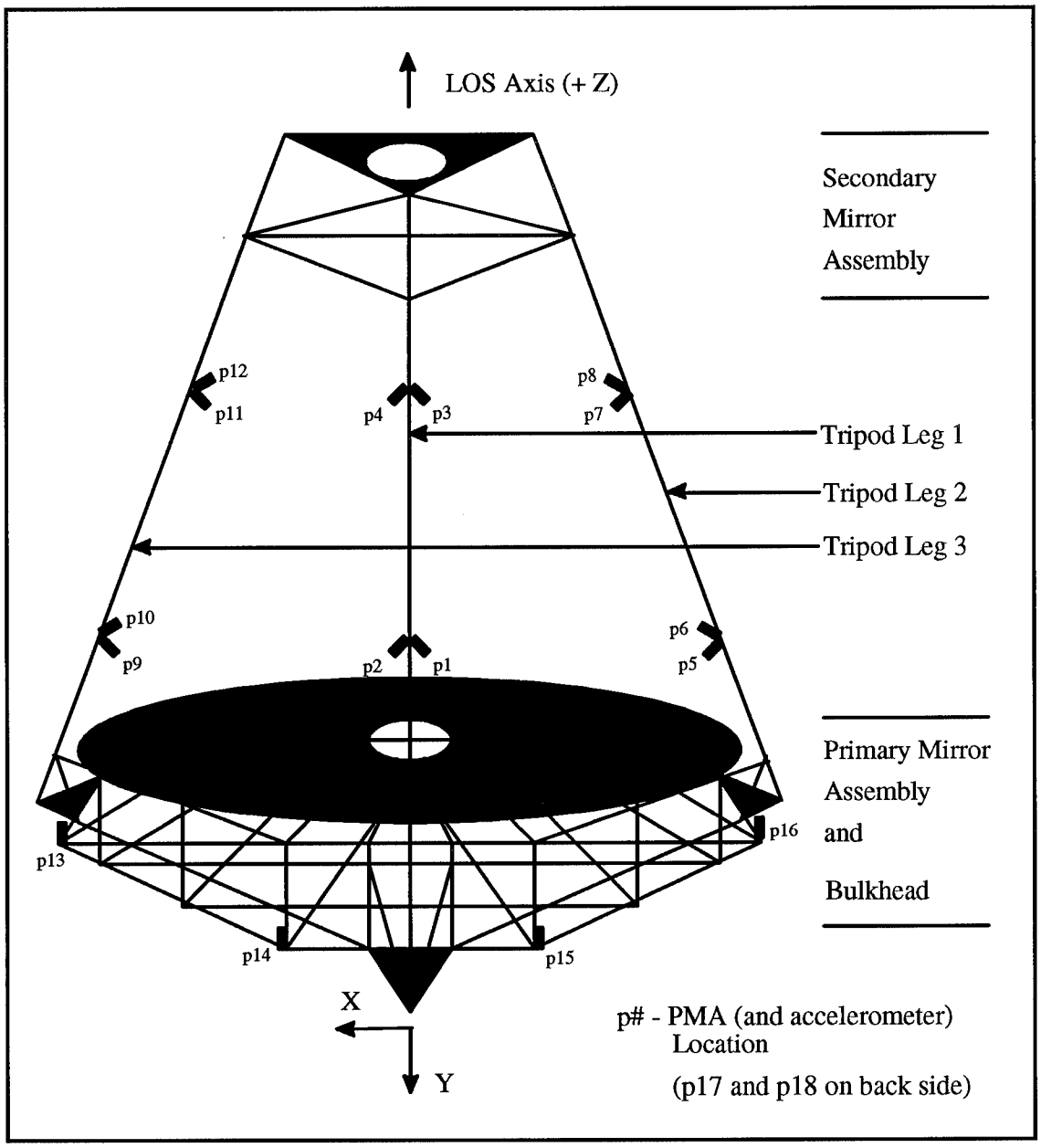


Figure 3-1. SPICE Structure [15:III-14]

meters. The Z-axis corresponds to the LOS axis and the Y axis points out the tripod leg number one. Each of the PMAs has its own local coordinate frame [31:3-2].

Alignment of the secondary mirror assembly and the Bulkhead is the primary concern of this research. Reduction of the linear and angular displacement between the two assemblies is the purpose of the control system. An exaggerated example of the SPICE structure exhibiting this misalignment due to its flexible bending modes is

illustrated in Figure 3-2. Note that for the purposes of this research, a purely torsional displacement about the line of sight, or Z-axis, will not alter the alignment of the bulkhead and secondary mirror assemblies.

3.2.2 Actuators and Sensors

Actuators provide the control force necessary to quell the structural vibrations,

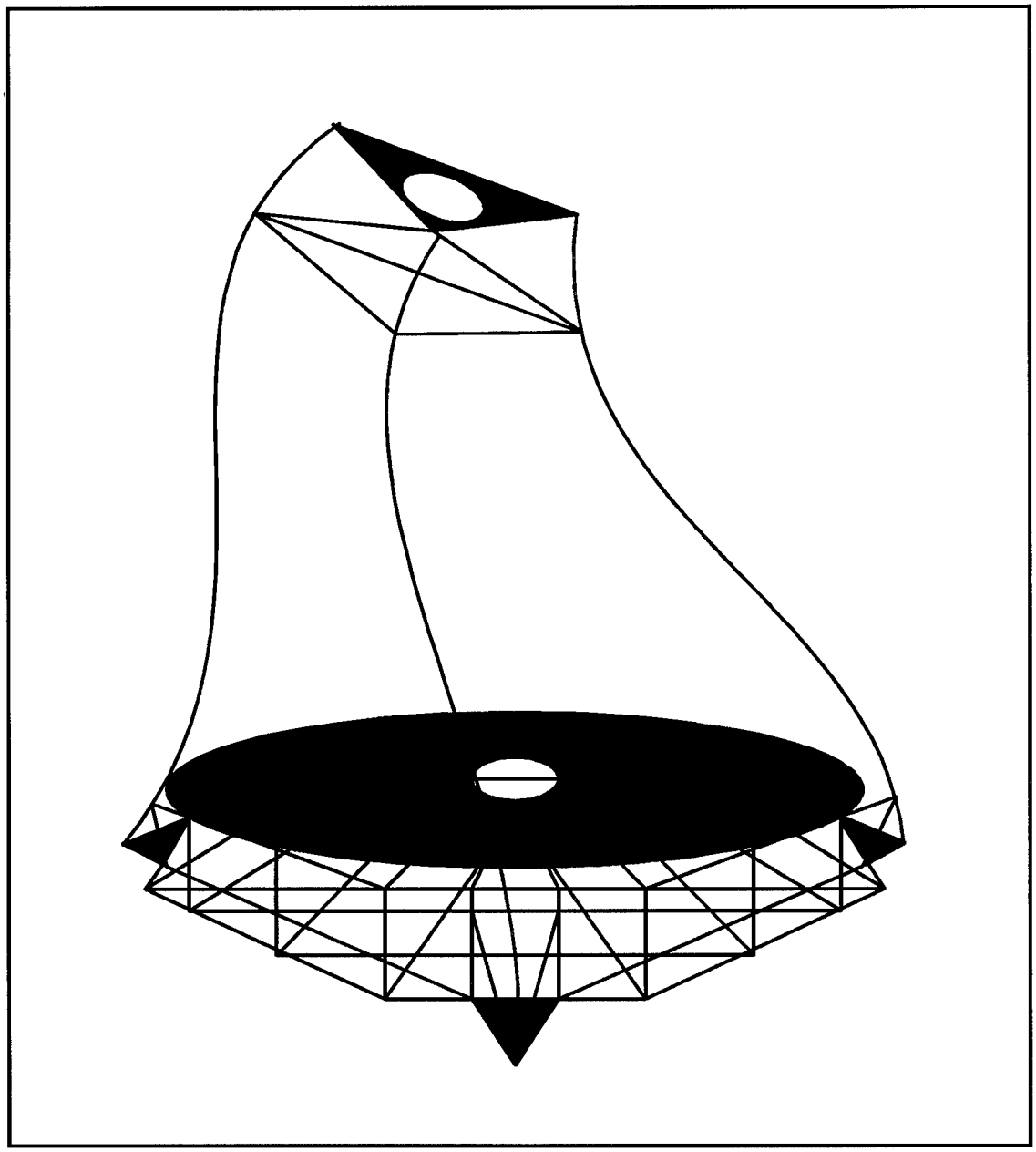


Figure 3-2. Flexible SPICE Structure

based on various sensor measurements. Sensor and actuator placement is an important part of the control system design [17:345]. The specific actuator utilized is referred to as a proof mass actuator (PMA). The PMA has a mass that is electro-magnetically moved to inhibit any bending motion of the structure at the location of the PMA. The dynamics of the PMA can be thought of as a simple spring-mass system [6:899]. A total of 18 PMAs are mounted on the structure. There are 6 PMAs located such that there is one at each of the hexagonal corners of the bulkhead pointing in the Z direction. The tripod legs house the remaining 12 PMAs, with each leg having two sets of PMAs mounted along the local orthogonal (one axis of which is aligned with the tripod leg) coordinate axis and located approximately one third and two thirds up the length of the leg respectively.

Various sensors provide measurement information for the control system. Specifically, three different types of sensors are being used. First, there are a total of 54 accelerometers separated into 18 sets of 3 (one set per PMA) which measure the bending motion of the structure. Each set contains 2 high frequency Wilcoxin accelerometers and 1 low frequency Sundestrand accelerometer. A high frequency accelerometer is physically mounted on each PMA proof mass, while the remaining two accelerometers are located on the physical structure at the point of attachment and along the reference axis of each PMA. The second type of sensor is the Linear Variable Differential Transformer (LVDT), which provides a differential position measurement of the PMA proof mass with respect to the structure. The third type of sensors are the elements of the Optical Scoring System (OSS) which uses lasers to provide line of sight (LOS) measurements between the two mirror assemblies.

3.2.3 Disturbances

Two different types of disturbances can affect the physical structure. First, two highly correlated disturbances can enter each tripod leg at the attachment to the bulkhead. The second type of disturbance, one per leg, enters each tripod leg at the SM Assembly.

These input points are chosen due to the ease of entering oscillations at these locations during on-ground testing.

3.3 System Mathematical Model Description

A mathematical model of the components of the SPICE-4 system will be presented. The overall system model in block diagram form is shown in Figure 3-3. The

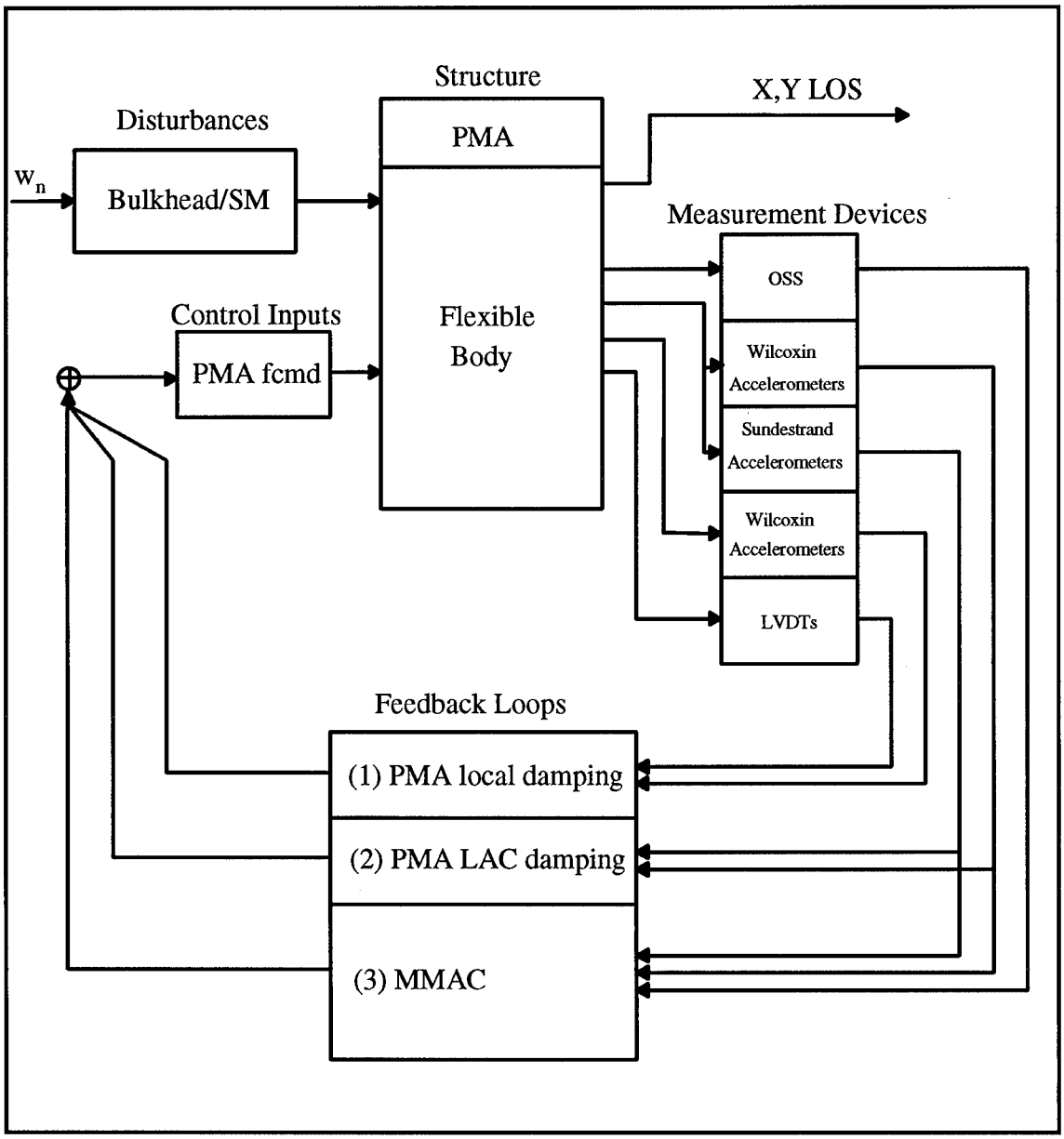


Figure 3-3. System Model High Level Block Diagram

actual full system model is composed of an extremely high number of states (~1000). The complete model is presented to provide insight and understanding of the actual physical system. In Figure 3-3, the PMA fcmd (force command) refers to the control inputs to the PMAs from the combined feedback loops. The PMA LAC (low authority control) damping refers to simple rate feedback for the structure, which introduces a moderate amount of damping to an otherwise very lightly damped structure [37:170], whereas the PMA local damping refers to localized damping for the PMAs. The individual parts of the overall system model will be further broken down and analyzed, thus providing a clearer understanding of the system components illustrated in Figure 3-3. Due to the extremely large number of states, the full system model cannot be used as a "truth" model because of computer limitations. In Section 3.4, the actual truth system model (reduced-order) will be presented with all the underlying assumptions and justifications.

3.3.1 Disturbances

The shaping filters of the noise models for all nine disturbance inputs to the structure are of the form of a fourth-order bandpass filter, with the only differences being gain changes for the correlated noises. This is depicted in Figure 3-4 (in this figure and subsequent figures, the numbers under the arrows specify the dimensions of the vector quantity). The driving noise for the shaping filter consists of six scalar white noises with equivalent statistics (denoted as the vector $\mathbf{w}_n$ in Figure 3-4). The noise strengths are such that the structure achieves a 100 micro-radian open loop RMS LOS error. The form of the state equation is given by [31:3-8]:

$$\dot{\mathbf{x}}_n(t) = \mathbf{F}_n\mathbf{x}_n(t) + \mathbf{G}_n\mathbf{w}_n(t) \tag{3.1}$$

where:

- $\mathbf{x}_n(t) =$ 24-state vector representing the disturbance states
- $\mathbf{F}_n =$ 24-by-24 constant fundamental dynamics matrix

- $\mathbf{G}_n =$ 24-by-6 constant noise input matrix
- $\mathbf{w}_n(t) =$ 6-by-1 unit-strength zero-mean white Gaussian noise vector

and the corresponding output equation is:

$$\mathbf{n}(t) = \mathbf{C}_n \mathbf{x}_n(t) \tag{3.2}$$

where:

- $\mathbf{n}(t)$ = 9-by-1 output colored noise vector
- $\mathbf{C}_n$ = 9-by-24 constant matrix

This filter contributes 24 states to the overall system model.

The output vector is defined as:

$$\mathbf{n} = \begin{bmatrix} n_1 \\ n_2 \\ n_3 \\ n_4 \\ n_5 \\ n_6 \\ n_7 \\ n_8 \\ n_9 \end{bmatrix} = \begin{bmatrix} \text{SMA disturbance Leg 1} \\ \text{SMA disturbance Leg 2} \\ \text{SMA disturbance Leg 3} \\ \text{Bulkhead disturbance 1 Leg 1} \\ \text{Bulkhead disturbance 1 Leg 2} \\ \text{Bulkhead disturbance 1 Leg 3} \\ \text{Bulkhead disturbance 2 Leg 1} \\ \text{Bulkhead disturbance2 Leg 2} \\ \text{Bulkhead disturbance 2 Leg 3} \end{bmatrix} \tag{3.3}$$

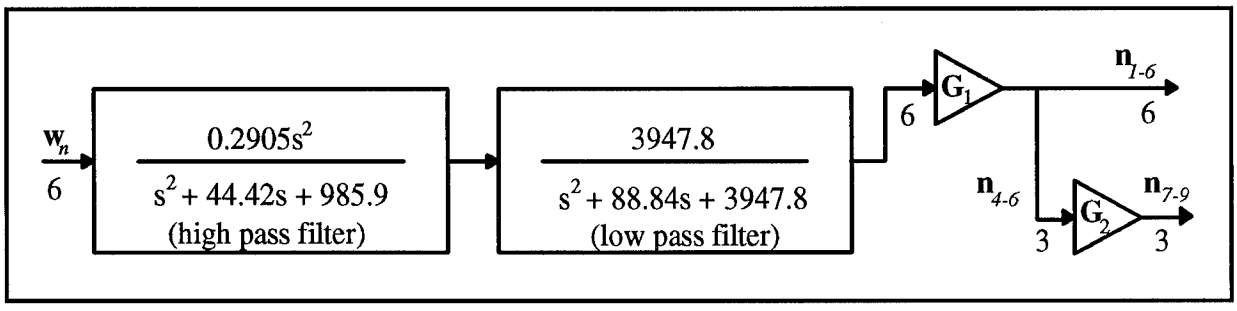


Figure 3-4. System Model Disturbance Block

3.3.2 Structure

The structure model refers to the dynamics of the flexible bending modes of the system and of the active control generated by the PMAs. The full-order flexible body consists of over 180 modes (n~180), with natural frequencies ranging from 7 to 150 Hz. This adds over 360 states to the overall system model since a position and a velocity state is inherent in the description of each mode. The system models for the PMAs are incorporated into the structure block simply by augmentation. Since the PMAs are essentially spring-mass systems, they can be modeled as simple second-order systems, each with a damping ratio of ~0.01 and natural frequency of ~5 Hz. This adds 36 states to the overall system model [31:3-10].

The system models were provided in the modal coordinate form; thus, no transformation from a physical representation was necessary (as discussed in Section 2.7.2). The structure block can be represented by the following state equation:

$$\dot{\mathbf{x}}_s(t) = \mathbf{F}_s\mathbf{x}_s(t) + \mathbf{B}_s\mathbf{u}_{fcmd}(t) + \mathbf{G}_s\mathbf{n}(t) \tag{3.4}$$

where:

- $\mathbf{x}_s(t)$ = (36 +2n)-state vector representing the flexible body and PMA modes
- $\mathbf{F}_s$ = (36+2n)-by-(36+2n) constant structure plant matrix
- $\mathbf{B}_s$ = (36+2n)-by-36 constant control input matrix
- $\mathbf{G}_s$ = (36+2n)-by-9 constant noise input matrix
- $\mathbf{n}(t)$ = defined in Equations (3.2) and (3.3)
- $\mathbf{u}_{fcmd}$ = 36-by-1 PMA force commands vector
- n = number of modes representing the structure

The actual structure of the dynamics matrix $\mathbf{F}_s$ is of the block diagonal form illustrated in Equation (2.79), where the corresponding state vector has the velocity states ordered first and the position states second. $\mathbf{x}_s(t)$ is shown on the following page:

$$\mathbf{x}_s(t) = \begin{bmatrix} \mathbf{x}_1 \\ \vdots \\ \mathbf{x}_{18} \\ \mathbf{x}_{18+1} \\ \vdots \\ \mathbf{x}_{18+n} \\ \mathbf{x}_{18+n+1} \\ \vdots \\ \mathbf{x}_{18+n+18} \\ \mathbf{x}_{36+n+1} \\ \vdots \\ \mathbf{x}_{36+2n} \end{bmatrix} = \begin{bmatrix} \text{PMA 1 velocity} \\ \vdots \\ \text{PMA 18 velocity} \\ \text{First bending mode velocity} \\ \vdots \\ \text{nth bending mode velocity} \\ \text{PMA 1 position} \\ \vdots \\ \text{PMA 18 position} \\ \text{First bending mode position} \\ \vdots \\ \text{nth bending mode position} \end{bmatrix} \tag{3.5}$$

The associated output equation is as follows:

$$\mathbf{y}_{struct}(t) = \mathbf{C}_s\mathbf{x}_s(t) + \mathbf{D}_{su}\mathbf{u}_{fcmd}(t) + \mathbf{D}_{sn}\mathbf{n}(t) \tag{3.6}$$

where:

- $\mathbf{y}_{struct}(t)$ = 98-by-1 structure output response vector
- $\mathbf{C}_s$ = 98-by-(36+2n) constant matrix
- $\mathbf{D}_{su}$ =98-by-36 deterministic control input direct feedthrough matrix
- $\mathbf{D}_{sn}$ = 98-by-9 noise direct feedthrough matrix

The output vector is defined as:

$$\mathbf{y} = \begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ \vdots \\ y_{44} \\ y_{45} \\ \vdots \\ y_{62} \\ y_{63} \\ \vdots \\ y_{80} \\ y_{81} \\ \vdots \\ y_{98} \end{bmatrix} = \begin{bmatrix} \text{X Line of Sight} \\ \text{Y Line of Sight} \\ \text{LOS sensor 1 (OSS1)} \\ \vdots \\ \text{LOS sensor 42 (OSS42)} \\ \text{Structure acceleration 1} \\ \vdots \\ \text{Structure acceleration 18} \\ \text{PMA acceleration 1} \\ \vdots \\ \text{PMA acceleration 18} \\ \text{Differential position 1 (PM1 wrt structure)} \\ \vdots \\ \text{Differential position 18 (PM18 wrt structure)} \end{bmatrix} \tag{3.7}$$

The X and Y Line of Sight outputs result from a transformation of the outputs of the optical scoring sensor or OSS outputs. The OSS outputs are position measurements of the displacement of the primary mirror assembly with respect to the secondary mirror assembly, as measured by a laser/sensor pair. The structural acceleration outputs relate the acceleration of the structure at the point of attachment of the PMAs. The PMA acceleration outputs relate the acceleration of the actual proof mass within each PMA. The differential position outputs relate the position of the proof mass relative to the structure at the point of attachment [31:3-12].

3.3.3 Measurement Devices

There are four different types of measurement devices included on the SPICE-4 structure. These include Wilcoxin high frequency accelerometers, Sundestrand low frequency accelerometers, LVDTs, and optical scoring sensors. Each type of sensor has its own system model.

3.3.3.1 Wilcoxin Accelerometers

The high frequency Wilcoxin accelerometer can be modeled by a second-order high-pass filter with correlated sensor noise as illustrated in Figure 3-5. The colored sensor noise is modeled with a high-pass third-order shaping filter. The state space representation is given by:

$$\dot{\mathbf{x}}_{Wacc}(t) = \begin{bmatrix} \dot{\mathbf{x}}_{wa}(t) \\ \dot{\mathbf{x}}_{nwa}(t) \end{bmatrix} = \begin{bmatrix} \mathbf{F}_{wa} & \mathbf{0} \\ \mathbf{0} & \mathbf{F}_{nwa} \end{bmatrix} \begin{bmatrix} \mathbf{x}_{wa}(t) \\ \mathbf{x}_{nwa}(t) \end{bmatrix} + \begin{bmatrix} \mathbf{B}_{wa} \\ \mathbf{0} \end{bmatrix} \mathbf{y}_{acc}(t) + \begin{bmatrix} \mathbf{0} \\ \mathbf{G}_{nwa} \end{bmatrix} \mathbf{w}_{Wacc}(t) \quad (3.8)$$

where:

- $\mathbf{x}_{wa}(t)$ = 36-state vector representing the accelerometer response
- $\mathbf{x}_{nwa}(t)$ = 54-state vector representing the time-correlated accelerometer noise
- $\mathbf{F}_{wa}$ = 36-by-36 constant accelerometer plant matrix
- $\mathbf{F}_{nwa}$ = 54-by-54 constant accelerometer noise shaping filter system matrix
- $\mathbf{B}_{wa}$ = 36-by-18 constant matrix
- $\mathbf{G}_{nwa}$ = 54-by-18 constant matrix
- $\mathbf{y}_{acc}$ = defined by the appropriate partition of Equation (3.7), (see below)
- $\mathbf{w}_{Wacc}$ = 18-by-1 unit-strength white Gaussian noise vector

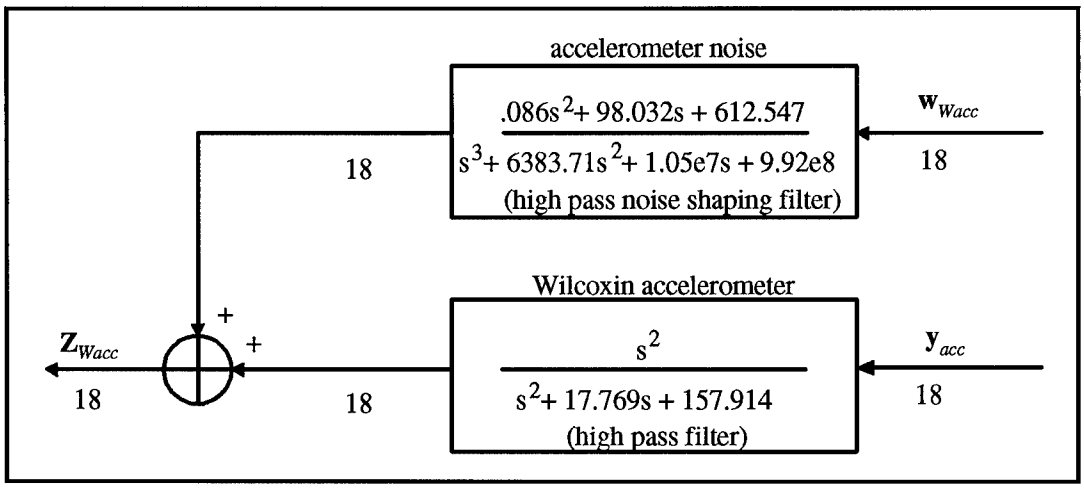


Figure 3-5. Wilcoxin Accelerometer Model

The corresponding measurement equation is given as:

$$\mathbf{z}_{Wacc}(t_i) = \begin{bmatrix} \mathbf{H}_{wa} & \mathbf{H}_{nwa} \end{bmatrix} \begin{bmatrix} \mathbf{x}_{wa}(t_i) \\ \mathbf{x}_{nwa}(t_i) \end{bmatrix} + \begin{bmatrix} \mathbf{D}_{ywa} \end{bmatrix} \mathbf{y}_{acc}(t_i) \tag{3.9}$$

where:

- $\mathbf{H}_{wa}$ = 18-by-36 constant accelerometer measurement matrix
- $\mathbf{H}_{nwa}$ = 18-by-54 constant accelerometer noise measurement matrix
- $\mathbf{D}_{ywa}$ = 18-by-18 constant feedthrough matrix
- $\mathbf{y}_{acc}$ = defined by the appropriate partition of Equation (3.7), (see below)

Since there are actually two sets of Wilcoxin accelerometers (located on the PMA and structure respectively), each represented by a separate model of the form given by Equations (3.8) and (3.9), the separate accelerometer input terms for the 18-dimensional $\mathbf{y}_{acc}$ vector are gained by extracting the appropriate set of components from Equation (3.7) (terms $\mathbf{y}_{63}$ through $\mathbf{y}_{80}$ for the PMA accelerometers and terms $\mathbf{y}_{45}$ through $\mathbf{y}_{62}$ for the structure accelerometers, respectively) [31:3-14].

3.3.3.2 Sundestrand Accelerometers

The low frequency Sundestrand accelerometer is appropriately modeled by a second-order low-pass filter as illustrated in Figure 3-6. The colored sensor noise is modeled with a second-order low-pass shaping filter. The state space representation is given by:

$$\dot{\mathbf{x}}_{Sacc}(t) = \begin{bmatrix} \dot{\mathbf{x}}_{sa}(t) \\ \dot{\mathbf{x}}_{nsa}(t) \end{bmatrix} = \begin{bmatrix} \mathbf{F}_{sa} & \mathbf{0} \\ \mathbf{0} & \mathbf{F}_{nsa} \end{bmatrix} \begin{bmatrix} \mathbf{x}_{sa}(t) \\ \mathbf{x}_{nsa}(t) \end{bmatrix} + \begin{bmatrix} \mathbf{B}_{sa} \\ \mathbf{0} \end{bmatrix} \mathbf{y}_{acc}(t) + \begin{bmatrix} \mathbf{0} \\ \mathbf{G}_{nsa} \end{bmatrix} \mathbf{w}_{Sacc}(t) \tag{3.10}$$

where:

- $\mathbf{x}_{sa}(t)$ = 36-state vector representing the accelerometer response
- $\mathbf{x}_{nsa}(t)$ = 36-state vector representing the time-correlated accelerometer noise
- $\mathbf{F}_{sa}$ = 36-by-36 constant accelerometer plant matrix
- $\mathbf{F}_{nsa}$ = 36-by-36 constant accelerometer noise shaping filter system matrix
- $\mathbf{B}_{sa}$ = 36-by-18 constant matrix
- $\mathbf{G}_{nsa}$ = 36-by-18 constant matrix
- $\mathbf{y}_{acc}$ = defined by Equation (3.7)
- $\mathbf{w}_{Sacc}$ = 18-by-1 unit-strength white Gaussian noise vector

The corresponding measurement equation is given as:

$$\mathbf{z}_{Sacc}(t_i) = \begin{bmatrix} \mathbf{H}_{sa} & \mathbf{H}_{nsa} \end{bmatrix} \begin{bmatrix} \mathbf{x}_{sa}(t_i) \\ \mathbf{x}_{nsa}(t_i) \end{bmatrix} \tag{3.11}$$

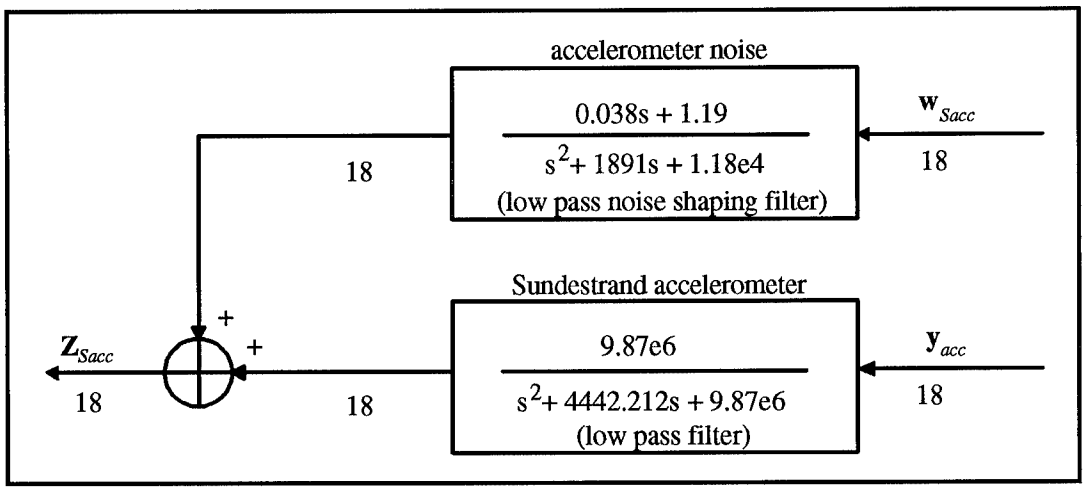


Figure 3-6. Sundestrand Accelerometer Model

where:

- $\mathbf{H}_{sa}$ = 18-by-36 constant accelerometer measurement matrix
- $\mathbf{H}_{nsa}$ = 18-by-36 constant accelerometer noise measurement matrix

The accelerometer input vector terms for $\mathbf{y}_{acc}$ are gained by extracting the appropriate components from Equation (3.7) (terms $\mathbf{y}_{45}$ through $\mathbf{y}_{62}$) [31:3-15].

3.3.3.3 LVDTs

The LVDT is modeled by a second-order low-pass filter as illustrated in Figure 3-7. The colored sensor noise is modeled with a second-order low-pass shaping filter. The state space representation is given by:

$$\dot{\mathbf{x}}_{LVDT}(t) = \begin{bmatrix} \mathbf{F}_{LVDT} & \mathbf{0} \\ \mathbf{0} & \mathbf{F}_{nLVDT} \end{bmatrix} \begin{bmatrix} \mathbf{x}_{LVDT}(t) \\ \mathbf{x}_{nLVDT}(t) \end{bmatrix} + \begin{bmatrix} \mathbf{B}_{LVDT} \\ \mathbf{0} \end{bmatrix} \mathbf{y}_{diff}(t) + \begin{bmatrix} \mathbf{0} \\ \mathbf{G}_{nLVDT} \end{bmatrix} \mathbf{w}_{LVDT}(t) \quad (3.12)$$

where:

- $\mathbf{x}_{LVDT}(t)$ = 36-state vector representing the LVDT response
- $\mathbf{x}_{nLVDT}(t)$ = 36-state vector representing the time-correlated LVDT noise

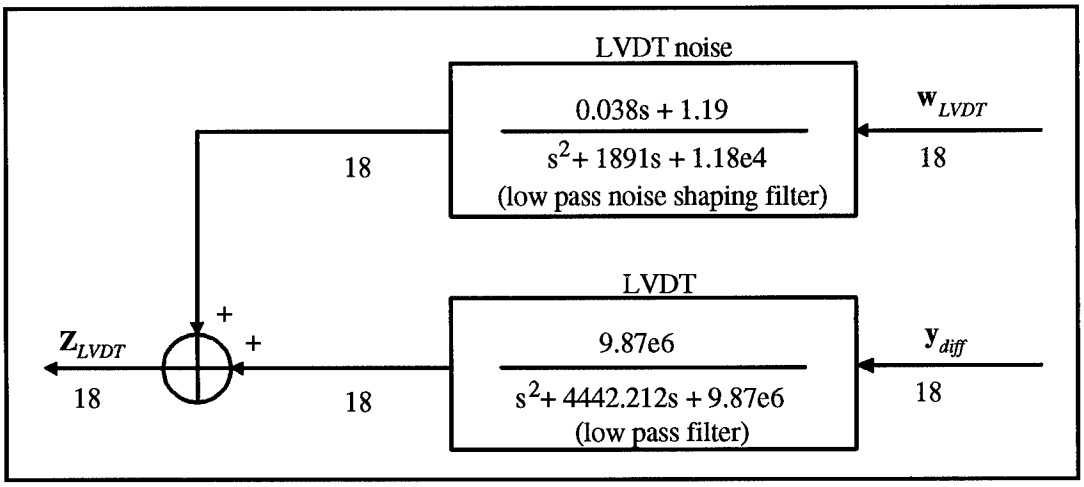


Figure 3-7. LVDT Model

- $\mathbf{F}_{LVDT}$ = 36-by-36 constant LVDT plant matrix
- $\mathbf{F}_{nLVDT}$ = 36-by-36 constant LVDT noise shaping filter system matrix
- $\mathbf{B}_{LVDT}$ = 36-by-18 constant matrix
- $\mathbf{G}_{nLVDT}$ = 36-by-18 constant matrix
- $\mathbf{y}_{diff}$ = defined by Equation (3.7)
- $\mathbf{w}_{LVDT}$ = 18-by-1 unit-strength white Gaussian noise vector

The corresponding measurement equation is given as:

$$\mathbf{z}_{LVDT}(t_i) = \begin{bmatrix} \mathbf{H}_{LVDT} & \mathbf{H}_{nLVDT} \end{bmatrix} \begin{bmatrix} \mathbf{x}_{LVDT}(t_i) \\ \mathbf{x}_{nLVDT}(t_i) \end{bmatrix} \tag{3.13}$$

where:

- $\mathbf{H}_{LVDT}$ = 18-by-36 constant LVDT measurement matrix
- $\mathbf{H}_{nLVDT}$ = 18-by-36 constant LVDT noise measurement matrix

The LVDT input vector terms for $\mathbf{y}_{diff}$ are gained by extracting the appropriate components from Equation (3.7) (terms $\mathbf{y}_{81}$ through $\mathbf{y}_{98}$) [31:3-16].

3.3.3.4 Optical Scoring Sensors

The OSS model is simply colored sensor noise which is modeled with a fourth-order band-pass shaping filter. The state space representation for the model depicted in Figure 3-8 is given by:

$$\dot{\mathbf{x}}_{nOSS}(t) = \begin{bmatrix} \mathbf{F}_{nOSS} \end{bmatrix} \mathbf{x}_{nOSS}(t) + \begin{bmatrix} \mathbf{G}_{nOSS} \end{bmatrix} \mathbf{w}_{OSS}(t) \tag{3.14}$$

where:

- $\mathbf{x}_{nOSS}(t)$ = 72-state vector representing the time correlated OSS noise
- $\mathbf{F}_{nOSS}$ = 72-by-72 constant OSS noise plant matrix
- $\mathbf{G}_{nOSS}$ = 72-by-18 constant matrix
- $\mathbf{w}_{OSS}$ = 18-by-1 unit-strength white Gaussian noise vector

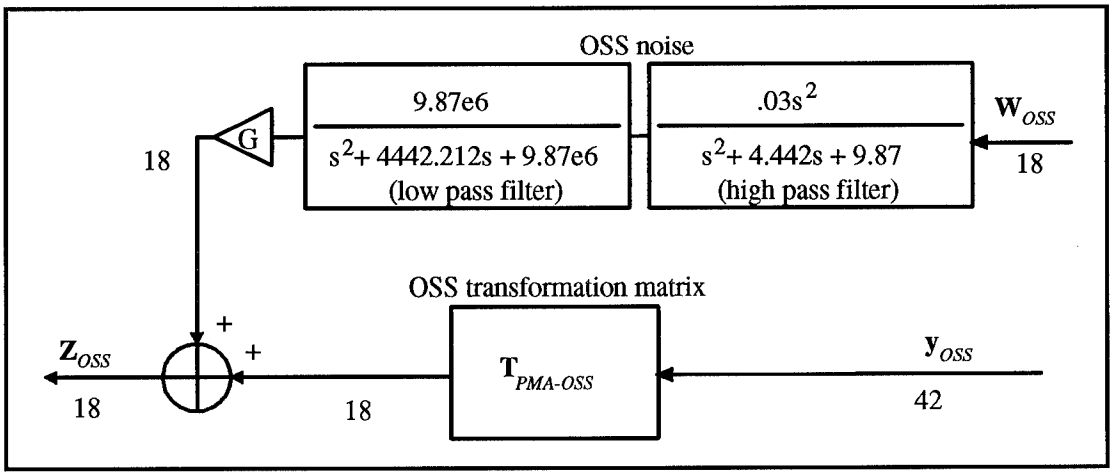


Figure 3-8. OSS Model

The 42 sensor outputs are transformed from the sensor axes to correspond to the 18 PMA nodes via a coordinate frame transformation, $\mathbf{T}_{PMA-OSS}$. Thus the form of the measurement equation is as follows:

$$\mathbf{z}_{OSS}(t_i) = [\mathbf{H}_{OSS}]\mathbf{x}_{nOSS} + [\mathbf{T}_{PMA-OSS}]\mathbf{y}_{OSS} \tag{3.15}$$

where:

- $\mathbf{H}_{OSS}$ = 18-by-72 constant matrix
- $\mathbf{T}_{PMA-OSS}$ = 18-by-42 constant OSS Transformation matrix

The OSS input vector terms for $\mathbf{y}_{OSS}$ are gained by extracting the appropriate components from Equation (3.7) (terms $\mathbf{y}_3$ through $\mathbf{y}_{44}$) [31:3-17].

3.3.4 Feedback Loops and Control Inputs

The three-tiered feedback loop structure was illustrated in Figure 3-3. The first tier, PMA local damping force, provides for localized damping feedback for the PMAs. The second tier, PMA Low Authority Control (LAC), provides simple rate feedback for the entire structure. The third tier is the MMAC design which provides feedback for attaining the desired performance specifications. Note that only the outputs of the

structure accelerometers and OSS are used by the MMAC loop. The PMA local damping and PMA LAC controllers are comprised of numerous integrators, differentiators, and filters which add an additional 216 states to the truth model. The PMA force command input block is modeled as two consecutive second order filters with a fourth order additive colored noise. A more complete illustration with all the appropriate models is provided in Figure 3-9 [31:3-18].

3.4 Truth Model Selection

The total full-order truth model consists of (690+2n) states. Driving noise contributes 24 states; structural bending modes and PMAs contribute (36+2n) states. The Wilcoxin accelerometers constitute 108 states, while the Sundestrand accelerometers contribute 72 states to the truth model. The LVDTs and the OSS sensors each contribute 72 states. PMA input noise is comprised of 144 additional states. Finally, the tiered feedback structure contributes 180 states, 72 for PMA local damping, and 108 for the PMA low authority controller.

This extremely large number of states presents a significant problem of computational burden in the controller development. Thus, reducing the total number of states in the truth model becomes necessary. However, the reduced truth model must still maintain a true representation of the physical structure. There are several assumptions made in performing the truth model reduction. First, the flexible bending modes are truncated at natural frequencies greater than 100 Hz. Second, the measurement devices are modeled as providing perfect measurements with additive white sensor noise. Third, any low-pass filter with a break frequency beyond the frequency range under consideration can be eliminated. Finally, noise inputs on the first two feedback loops and on the PMA control inputs will be eliminated. Justifications for each of these methods are addressed in the following section [31:3-20].

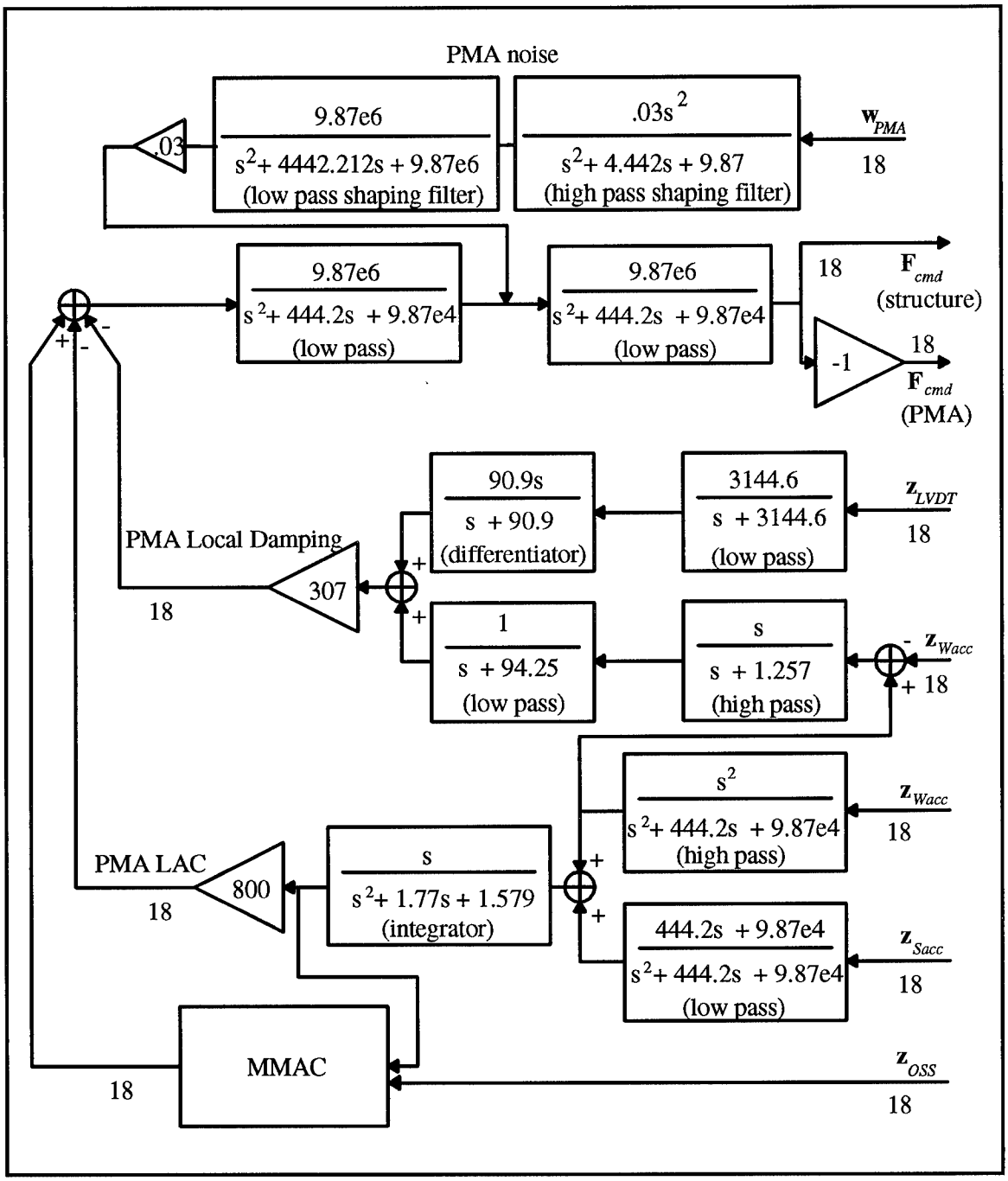


Figure 3-9. Feedback Loops and Control Inputs Model

3.4.1 Truth Model Simplifications

Truncation of the system model bending modes at 100 Hz will eliminate 86 flexible body modes, or 172 states, resulting in 216 bending mode states in the truth model. The

underlying assumption is that any mode at higher frequencies will be essentially instantaneously quelled thus having no effect on the system. Moreover, the amplitude associated with these modes are significantly dominated by the amplitudes of low frequency mode effects. A similar assumption was made by Gustafson in his work with the SPICE-2 model [7]. The resulting frequency range (5 - 100 Hz) forms the total frequency range of interest to be considered in this research.

A large number of overall system states are contributed by the sensor, noise, and feedback loop models. Many of these models can be eliminated based on the assumption that all measurements have additive *white* noise. This assumption is justified as follows.

In the MMAC feedback, three sensors are utilized: the Sundestrand accelerometer, the Wilcoxin accelerometer, and the OSS. Viewed together, the combination of the high frequency Wilcoxin accelerometer and the low frequency Sundestrand model can be treated as a unity gain bandpass over the entire frequency range of interest. Thus, they can be thought of as providing "perfect" (though noise-corrupted) measurements of acceleration (i.e., having a flat Bode amplitude ratio plot for the transfer function from true to sensed acceleration) since both cut-off frequencies are outside the range of interest. There is no attenuation in the frequency range of interest. Thus the states associated with these accelerometers can be eliminated. This same situation applies to the colored noise added to each of these respective sensors. The Sundestrand noise is shaped by a low-pass filter and the Wilcoxin noise is shaped by a high-pass filter. Thus their replacement by white noise is justified for the same reasons as above. The additional filters in this loop are no longer necessary and thus they are eliminated. The colored OSS sensor noise is formed by a band-pass filter where the high and low break frequencies are outside the frequency range of interest, and thus this noise can be treated as white.

In the physical system, measurements of acceleration and position are provided. These must be integrated and differentiated respectively to gain the required rate feedback.

However, in the mathematical representation, direct "velocity measurements" can be obtained for each of the sensor outputs. Therefore, the integrators and differentiators are no longer necessary and their states can be eliminated from the system model.

The two sequential low-pass filters on the input of the PMA force commands both have break frequencies well beyond the frequency range of interest. Thus, they are unneeded and are removed from the system model.

Finally, the noise inputs on the first two feedback loops and the PMA force commands (not including the disturbance noise inputs and the sensor noise on the inputs to the MMAC) are assumed to have negligible impact on the system performance since the disturbance noises are of much higher magnitude. Thus, these are eliminated for simplicity [31:3-22].

3.4.2 Truth Model

The final version of the truth model with 294 total states is shown in Figure 3-10. The disturbance input models have not been altered. The structure state equation remains unchanged except that the number of flexible bending modes has been reduced to 108 (n=108) by eliminating high frequency bending modes. The new associated output equation is given by:

$$\mathbf{y}_{struct}(t) = \mathbf{C}_s \mathbf{x}_s(t) \tag{3.16}$$

where:

- $\mathbf{y}_{struct}(t)$ = 56-by-1 structure output response vector
- $\mathbf{C}_s$ = 56-by-(36+2n) constant matrix

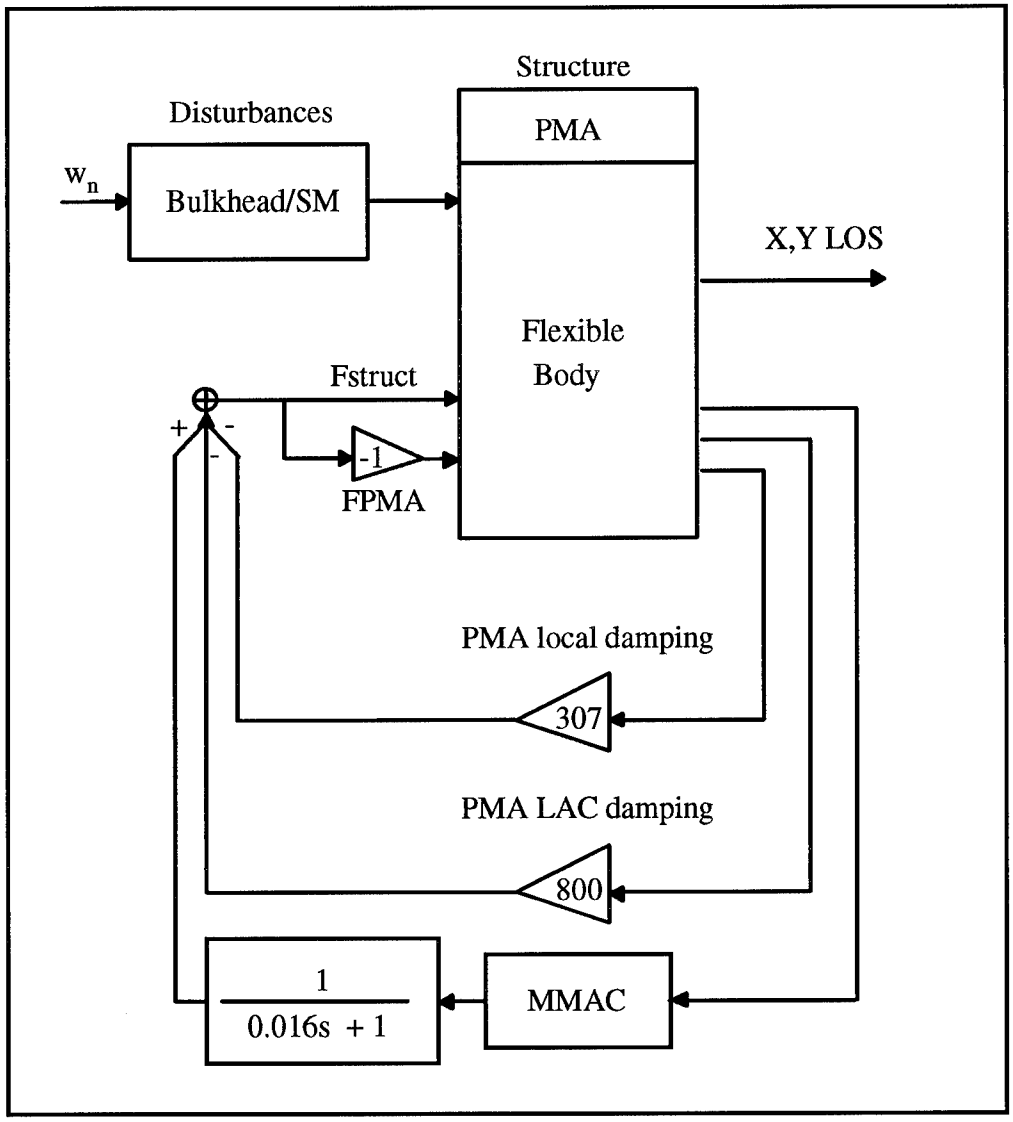


Figure 3-10. Truth Model Block Diagram

The $\mathbf{D}_{su}$ and $\mathbf{D}_{sn}$ terms from Equation (3.6) have been eliminated since the "velocity measurements" do not involve feedthrough terms. The output vector is now defined as:

$$\mathbf{y} = \begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ \vdots \\ y_{20} \\ y_{21} \\ \vdots \\ y_{38} \\ y_{39} \\ \vdots \\ y_{56} \end{bmatrix} = \begin{bmatrix} \text{X Line of Sight} \\ \text{Y Line of Sight} \\ \text{Transformed OSS 1} \\ \vdots \\ \text{Transformed OSS 18} \\ \text{Structure velocity 1} \\ \vdots \\ \text{Structure velocity 18} \\ \text{PMA velocity 1} \\ \vdots \\ \text{PMA velocity 18} \end{bmatrix} \tag{3.17}$$

where outputs $\mathbf{y}_{63-80}$ from Equation (3.7) have been eliminated. The velocity measurements for the PMA local damping loop are obtained by subtracting outputs $\mathbf{y}_{39-56}$ from outputs $\mathbf{y}_{20-38}$ respectively. Also, the transformation matrix for the OSS LOS sensors was incorporated into the output portion of the structure block, thus reducing the 42 individual sensor outputs to the 18 outputs to the MMAC block [31:3-26].

The addition of the first-order low-pass filter on the control inputs from the MMAC block produces a 10 Hz rolloff on the MMAC loop which is needed for stability robustness [30,31:3-26]. The state equation is given by:

$$\dot{\mathbf{x}}_f(t) = \mathbf{F}_f \mathbf{x}_f(t) + \mathbf{B}_f \mathbf{u}_{MMAC}(t) \tag{3.18}$$

where:

- $\mathbf{x}_f(t)$ = 18-state vector representing the filter states
- $\mathbf{F}_f$ = 18-by-18 constant filter plant matrix
- $\mathbf{B}_f$ = 18-by-18 constant matrix
- $\mathbf{u}_{MMAC}(t)$ = 18-by-1 MMAC control vector

No dynamics driving noise is necessary for this filter since it is implemented digitally and the states are known exactly. The corresponding output equation is:

$$\mathbf{f}_{MMAC}(t) = \mathbf{C}_f \mathbf{x}_f(t) \tag{3.19}$$

where:

- $\mathbf{f}_{MMAC}(t)$ = 18-by-1 output control vector
- $\mathbf{C}_f$ = 18-by-18 constant matrix

The output vector is defined as:

$$\mathbf{f}_{MMAC} = \begin{bmatrix} f_1 \\ \vdots \\ f_{18} \end{bmatrix} = \begin{bmatrix} \text{Filtered MMAC control to PMA1} \\ \vdots \\ \text{Filtered MMAC control to PMA18} \end{bmatrix} \tag{3.20}$$

where $\mathbf{f}_{MMAC}$ is then combined with the other feedback loop inputs to form the total input $\mathbf{f}_{cmd}$ to the structure block.

The complete open loop system model is formed by augmenting all the individual components together. The augmented state equation is given by:

$$\dot{\mathbf{x}}_{as}(t) = \mathbf{F}_{as}\mathbf{x}_{as}(t) + \mathbf{B}_{as}\mathbf{u}_{fcmd}(t) + \mathbf{G}_{as}\mathbf{w}(t) \tag{3.21}$$

where the augmented system matrices are given by:

$$\mathbf{F}_{as} = \begin{bmatrix} \mathbf{F}_{n(24x24)} & \mathbf{0}_{(24x18)} & \mathbf{0}_{(24x36+2n)} \\ \mathbf{0}_{(18x24)} & \mathbf{F}_{f(18x18)} & \mathbf{0}_{(18x36+2n)} \\ \mathbf{G}_s\mathbf{C}_{n(36+2nx24)} & \mathbf{B}_s\mathbf{C}_{f(36+2n\,x18)} & \mathbf{F}_{s(36+2n\,x36+2n)} \end{bmatrix}_{78+2n\,x78+2n} \tag{3.22}$$

$$\mathbf{B}_{as} = \begin{bmatrix} \mathbf{0}_{(24x18)} \\ \mathbf{B}_{f(18x18)} \\ \mathbf{0}_{(36+2n\,x18)} \end{bmatrix}_{78+2n\,x18} \tag{3.23}$$

$$\mathbf{G}_{as} = \begin{bmatrix} \mathbf{G}_{n(24x6)} \\ \mathbf{0}_{(18x6)} \\ \mathbf{0}_{(36+2n\,x6)} \end{bmatrix}_{78+2n\,x6} \tag{3.24}$$

and $\mathbf{w}(t)$ is still a white Gaussian noise vector with $\mathbf{Q}_t$ = the identity matrix. The augmented state vector is given by:

$$\mathbf{x}_{as}(t) = \begin{bmatrix} \mathbf{x}_1 \\ \vdots \\ \mathbf{x}_{24} \\ \mathbf{x}_{25} \\ \vdots \\ \mathbf{x}_{42} \\ \mathbf{x}_{43} \\ \vdots \\ \mathbf{x}_{60} \\ \mathbf{x}_{61} \\ \vdots \\ \mathbf{x}_{168} \\ \mathbf{x}_{169} \\ \vdots \\ \mathbf{x}_{186} \\ \mathbf{x}_{187} \\ \vdots \\ \mathbf{x}_{294} \end{bmatrix} = \begin{bmatrix} \text{1st disturbance state} \\ \vdots \\ \text{24th disturbance state} \\ \text{1st MMAC filter state} \\ \vdots \\ \text{18th MMAC filter state} \\ \text{PMA 1 velocity} \\ \vdots \\ \text{PMA 18 velocity} \\ \text{1st bending mode velocity} \\ \vdots \\ \text{nth bending mode velocity} \\ \text{PMA 1 position} \\ \vdots \\ \text{PMA 18 position} \\ \text{1st bending mode position} \\ \vdots \\ \text{nth bending mode position} \end{bmatrix} \tag{3.25}$$

The associated augmented output equation is given by:

$$\mathbf{y}_{as}(t) = \mathbf{C}_{as}\mathbf{x}_{as}(t) \tag{3.26}$$

where the augmented output matrix is given by:

$$\mathbf{C}_{as} = \begin{bmatrix} 0_{(56x24)} & 0_{(56x18)} & C_{s(56x36+2n)} \end{bmatrix} \tag{3.27}$$

and the output vector is still defined by Equation (3.17).

The PMA local damping and PMA LAC loops are simply formed by measurement feedback defined by the following:

$$\mathbf{z}_{PMAldamp}(t_i) = \begin{bmatrix} \mathbf{I}_{(18x18)} & -\mathbf{I}_{(18x18)} \end{bmatrix} \mathbf{y}_{as(21-38,39-56)}(t_i) \tag{3.28}$$

$$\mathbf{z}_{LAC}(t_i) = \begin{bmatrix} \mathbf{I}_{(18x18)} \end{bmatrix} \mathbf{y}_{as(21-38)}(t_i) \tag{3.29}$$

where the subscripted numbers for $\mathbf{y}_{as}$ (here and in subsequent equations) refer to the appropriate partitions from Equation (3.17). As mentioned is Section 3.4.1, no additional noise is added to these loops [31:3-28].

The measurements input to the MMAC design are defined by the following equation:

$$\mathbf{z}_{MMACinput}(t_i) = \begin{bmatrix} \mathbf{z}_{OSS}(t_i) \\ \mathbf{z}_{Svel}(t_i) \end{bmatrix} = \begin{bmatrix} \mathbf{I}_{(18x18)} & \mathbf{0} \\ \mathbf{0} & \mathbf{I}_{(18x18)} \end{bmatrix} \mathbf{y}_{as(3-38)}(t_i) + \mathbf{v}(t_i)_{(36x1)} \tag{3.30}$$

Substituting the appropriate partitions of Equation (3.27) into Equation (3.30) results in the format consistent with the Kalman filter measurement presentation given in Chapter 2:

$$\mathbf{z}_{MMACinput}(t_i) = \begin{bmatrix} \mathbf{z}_{OSS}(t_i) \\ \mathbf{z}_{Svel}(t_i) \end{bmatrix} = \begin{bmatrix} \mathbf{H}_{Cas(3-20)} \\ \mathbf{H}_{Cas(21-38)} \end{bmatrix} \mathbf{x}_{as}(t_i) + \mathbf{v}(t_i)_{(36x1)} \tag{3.31}$$

where:

- $\mathbf{z}_{MMACinput}$ = 36-by-1 measurement output vector
- $\mathbf{H}_{Cas(3-20)}$ = 18-by-36+2n constant measurement matrix
- $\mathbf{H}_{Cas(21-38)}$ = 18-by-36+2n constant measurement matrix
- $\mathbf{v}(t_i)$ = 36-by-1 white measurement noise vector

The individual matrices for the complete truth model can be found in [31].

3.5 Reduced-Order Filter Models

The primary area investigated for the order reduction efforts was the number of flexible body modes retained. Schiller [31] developed several reduced-order models based on both the modal reduction and the modal cost reduction techniques described in Chapter 2, with the goal of the reduction being to alleviate the computational loading.

For the modal reduction technique, three reduced-order models were developed. Table 3-1 lists the first 30 modes of the flexible body portion of the structure model. Several groupings were observed in the natural frequencies which provide the ideal locations to perform the modal reduction process. The three reduced-order models retained 12, 18, and 26 flexible body modes. Bode plots for the truth model and the three individual filter models were compared to show the good correspondence between all the filter plots and the truth plots at the lower frequencies.

For the modal cost reduction technique, four reduced-order models were developed. Table 3-2 contains the first largest 30 component cost values with the corresponding mode numbers. Once again, several natural grouping were observed (separated by relatively large gaps in the sequence). Three reduced-order models were developed, where the resulting number of flexible body modes was 10, 15, and 20, respectively. Additionally, a 26-mode model was subsequently created to allow direct comparison with the 26-mode model using the modal reduction technique. Note that the three lowest frequency modes are at the top of the modal cost table. This indicates that the lowest frequency modes are dominant in the structure's dynamic response. Bode plots for the modal cost filter models were compared to the truth model. Similar to the modal case, there is a good match between the reduced-order system model plots and the truth model plots at the lower frequencies.

The complete Bode plot comparisons as well as the filter model matrices for the 26-mode modal and modal cost reduced models can be found in the appendices to Schiller

[31]. Since Schiller had the most success with the 26-mode modally reduced models, they will be implemented for use in this research.

3.6 Summary

This chapter presented the models for the SPICE-4 structure used in this research. This included a physical description of the structure and each of its subsystems as well as a development of the full system mathematical models. Justification was provided for the development of a "reduced" truth model which consisted of 294 states and 108 flexible bending modes. The reduced-order filter models were described as containing 130 states and 26 flexible bending modes.

The following chapter will discuss the software and performance analysis used to simulate and evaluate the MMAE/MMAC for the SPICE-4 structure.

Mode	Eigenvalue	Damping Ratio	Natural Frequency (Hz)
1	-0.22 ± 43.29j	0.0102	6.889
2	-0.10 ± 49.00j	0.0045	7.800
3	-0.11 ± 49.40j	0.0047	7.8622
4	-0.31 ± 98.03j	0.0064	15.601
5	-0.33 ± 101.60j	0.0066	16.172
6	-0.47 ± 106.85j	0.0088	17.007
7	-0.35 ± 121.52j	0.0058	19.342
8	-0.91 ± 124.65j	0.0146	19.840
9	-0.56 ± 130.94j	0.0086	20.839
10	-0.37 ± 138.53j	0.0054	22.047
11	-1.24 ± 151.69j	0.0164	24.144
12	-0.57 ± 152.13j	0.0076	24.214
13	-1.39 ± 199.83j	0.0140	31.805
14	-1.45 ± 207.39j	0.0140	33.008
15	-1.49 ± 213.79j	0.0140	34.027
16	-1.53 ± 219.70j	0.0140	34.967
17	-1.55 ± 222.66j	0.0140	35.439
18	-1.58 ± 226.63j	0.0140	36.071
19	-1.92 ± 275.24j	0.0140	43.807
20	-1.99 ± 285.58j	0.0140	45.453
21	-2.01 ± 288.01j	0.0140	45.840
22	-2.09 ± 298.92j	0.0140	47.576
23	-2.16 ± 309.46j	0.0140	49.254
24	-2.18 ± 312.05j	0.0140	49.666
25	-2.19 ± 313.34j	0.0140	49.871
26	-2.21 ± 316.51j	0.0140	50.376
27	-2.25 ± 322.17j	0.0140	51.276
28	-2.28 ± 326.76j	0.0140	52.007
29	-2.31 ± 331.21j	0.0140	52.714
30	-2.56 ± 366.25j	0.0140	58.292

Table 3-1. Modal Eigenvalues and Natural Frequencies of the First 30 Modes

Component Cost ($x10^{-8}$)	Mode	Component Cost ($x10^{-8}$)	Mode	Component Cost ($x10^{-8}$)	Mode
0.348758	3	0.004097	7	0.001411	22
0.344947	2	0.003547	26	0.001166	35
0.031164	1	0.003536	24	0.001089	39
0.027831	4	0.003502	28	0.000880	34
0.012824	12	0.003343	23	0.000485	29
0.008589	8	0.002837	21	0.000434	30
0.006154	5	0.002753	6	0.000266	74
0.005481	9	0.002652	11	0.000264	70
0.005147	20	0.002610	25	0.000262	13
0.004712	27	0.002578	10	0.000256	99

Table 3-2. Top 30 Component Cost Values and Associated Modes

IV. Simulation

4.1 Introduction

This chapter will discuss the computer simulations and software which are used to analyze the performance of the multiple model adaptive estimator and controller as they are applied to the SPICE-4 system model. Monte Carlo analysis will be the method of gathering statistical information about the process. Simulations will be performed using software developed in previous research [2,3,7,9,10,13,29,31,32,35] but modified to meet the specific goals of this effort. The discretization of the parameter space will also be addressed in this chapter.

4.2 Monte Carlo Analysis

In order to obtain statistics of the MMAE/MMAC performance, a Monte Carlo analysis is performed. The simulation software will produce multiple-run sample statistics of the estimator and controller performance. The data of interest are estimation errors for the filter performance, line of sight errors for the controller performance, and parameter estimation errors (as important though secondary in nature) in both cases. A covariance analysis cannot be performed for this problem due to the adaptive nature of the MMAE/MMAC technique, which precludes analytical evaluation of second moments of the errors of interest [20:329].

As presented in Chapter 3, the main emphasis for this research will be to analyze an MMAE/MMAC algorithm (based on a 130-state reduced-order filter model) against a 294-state SPICE-4 truth model. The truth model contains all 108 flexible bending modes of the structure, whereas the filter design model contains only 26 bending modes. This filter model choice was based on the results of Schiller [31] who found that this modally reduced model performed best in the one-dimensional parameter space.

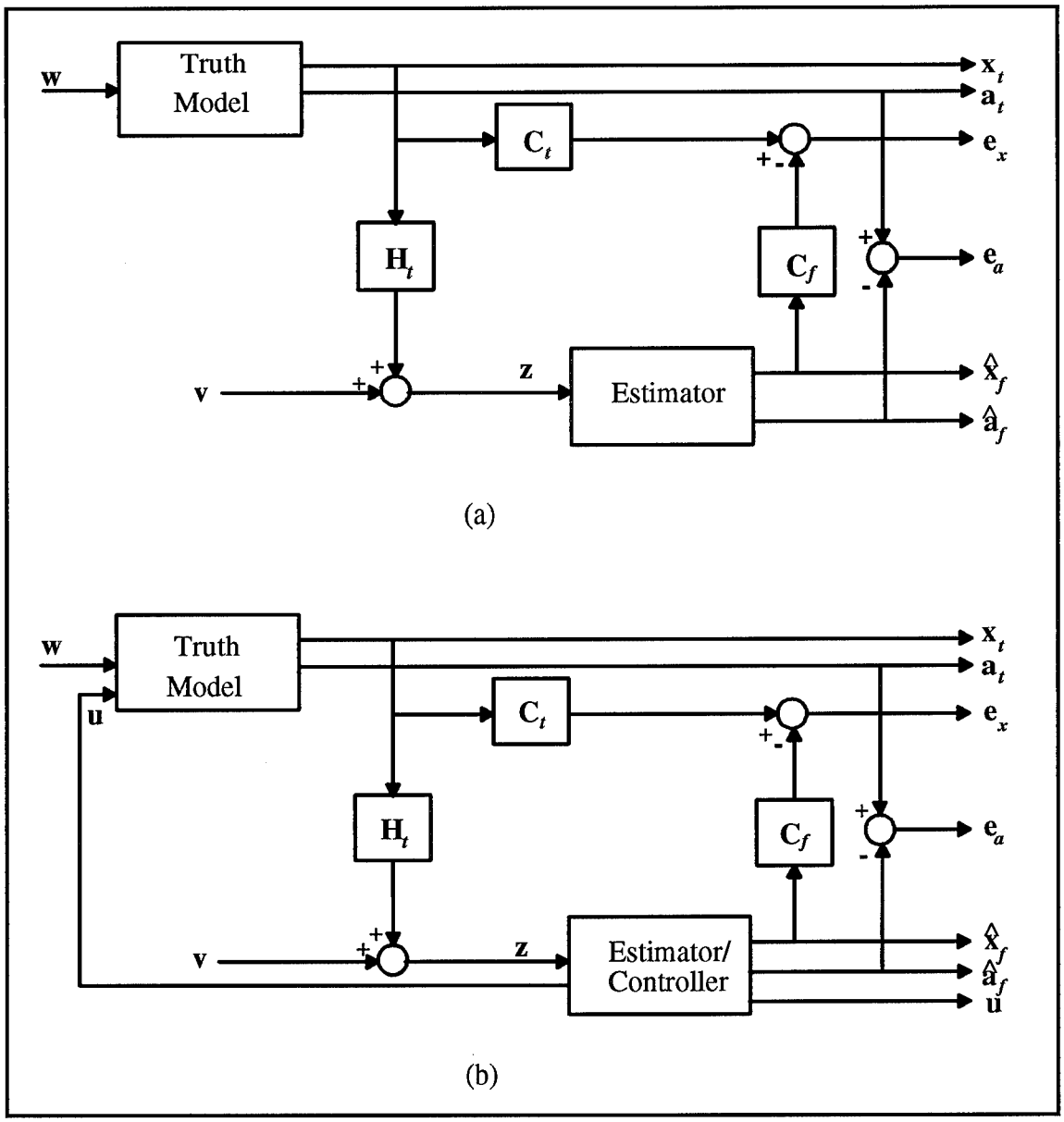


Figure 4-1. (a) Estimator Simulation, and (b) Controller Simulation [32]

The simulations can be broken into two parts: estimator analysis and controller analysis, as depicted in Figure 4-1. Figure 4-1a shows the estimator analysis and Figure 4-1b shows the controller analysis. The variables shown in Figure 4-1 are as follows:

- $\mathbf{x}_t(t_i)$ = the truth model states
- $\hat{\mathbf{x}}_f(t_i)$ = filter estimates of the system states

- $\mathbf{a}_t(t_i)$ = the uncertain parameter vector implemented in the truth model
- $\hat{\mathbf{a}}_f(t_i)$ = filter estimates of the uncertain parameter vector
- $\mathbf{e}_a(t_i)$ = the error in the parameter estimate defined as: $\mathbf{e}_a(t_i) = \mathbf{a}_t(t_i) - \hat{\mathbf{a}}_f(t_i)$
- $\mathbf{e}_x(t_i)$ = the error in the system estimate

The following sections will present the error vector formulation and the error vector statistics.

4.2.1 Error Vector Formulation

The main purpose of the controller is to quell vibrations in the SPICE-4 structure such that the line of sight (LOS) vector through the center of the structure is maintained precisely. Therefore, the X and Y axis line of sight (LOS) deviations will determine the performance of the reduced-order model based filter and controller algorithms. The error vector for filter performance evaluation is determined by subtracting the X and Y axis LOS values as generated by the filter from the corresponding truth model calculated values (Figure 4-1(a)). The equation to determine this error vector is given by:

$$\mathbf{e}_x(t_i) = \mathbf{C}_t \mathbf{x}_t(t_i) - \sum_{j=1}^{K} \mathbf{C}_{f_j} \hat{\mathbf{x}}_{f_j}(t_i) \cdot p_j(t_i) \tag{4.1}$$

where $\mathbf{C}_t$ and $\mathbf{C}_{f_j}$ are the output matrices used to determine the X and Y LOS deviations for the truth model and filter model respectively. The summation in Equation (4.1) is formed using the conditional probabilities in the MMAE implementation as presented in Chapter 2, where the subscript *f* represents filter and the index *j* indicates which of the moving bank filter's weighted estimates are being summed. The specific form of the error vector is given by:

$$\mathbf{e}_x(t_i) = \begin{bmatrix} e_{x_1}(t_i) \\ e_{x_2}(t_i) \end{bmatrix} = \begin{bmatrix} \text{X} - \text{axis LOS position error} \\ \text{Y} - \text{axis LOS position error} \end{bmatrix} \tag{4.2}$$

In Figure 4-1(b), the MMAC controller is implemented for closed loop estimation and control. The purpose of the controller is to drive the vibrations of the structure to zero, or in other words, regulate the position deviations. Thus, performance of the controller is indicated from an analysis of the actual X and Y-axis LOS deviations. This error vector is given by:

$$\mathbf{e}'_x(t_i) = \mathbf{C}_t \mathbf{x}_t(t_i) \tag{4.3}$$

where $\mathbf{e}'_x(t_i)$ is the vector of truth model LOS deviations.

4.2.2 Error Vector Statistics

Since the process is assumed to be Gaussian for this effort, the mean and covariance of the estimation error and LOS deviations are the main statistics used to determine performance. The mean is calculated by [13:74,35:67]:

$$E\{\mathbf{e}_x(t_i)\} \approx \mathbf{M}_{e_x}(t_i) = \frac{1}{L}\sum_{l=1}^{L} \mathbf{e}_{x_l}(t_i) \tag{4.4}$$

where L is the number of Monte Carlo runs made and $\mathbf{e}_{x_l}(t_i)$ is the value of the error signal during the l^{th} simulation at run time t_i. The covariance of the mean error signal is calculated by [20:130]:

$$\begin{aligned}\mathbf{P}_{e_x}(t_i) &= E\left\{\left[\mathbf{e}_x(t_i) - E\{\mathbf{e}_x(t_i)\}\right]\left[\mathbf{e}_x(t_i) - E\{\mathbf{e}_x(t_i)\}\right]^{\mathrm{T}}\right\} \\ &\approx \frac{1}{L-1}\sum_{l=1}^{L}\left\{\mathbf{e}_{x_l}(t_i)\mathbf{e}_{x_l}^{\mathrm{T}}(t_i)\right\} - \frac{L}{L-1}\mathbf{M}_{e_x}(t_i)\mathbf{M}_{e_x}^{\mathrm{T}}(t_i)\end{aligned} \tag{4.5}$$

The statistics for the parameter estimation errors, $\mathbf{e}_a(t_i)$, and the LOS deviations, $\mathbf{e}'_x(t_i)$, are obtained by using the above equations with the appropriate variables substituted. The statistics will be based on a ten-run Monte Carlo analysis, with each run having a duration of ten seconds. A ten-run analysis was chosen to keep the simulation times reasonable while producing sample statistics that reflect the true underlying statistics adequately.

The temporal average of the line of sight errors is useful for quantifying the controller performance in a compact manner. This value is calculated by the following:

$$e_{tp} = \frac{1}{N} \sum_{j=i-N+1}^{i} \sqrt{\left[\mathbf{M}_{e_x}(t_j) \mathbf{M}_{e_x}^{\mathrm{T}}(t_j) + \mathbf{P}_{e_x}(t_i) \right]_{pp}} \quad \text{for } p = 1 \text{ and } 2 \tag{4.6}$$

where e_{tp} is the temporal average of the p^{th} component of $\mathbf{e}_x$ or $\mathbf{e}'_x$ and N is the number of sample periods [7]. For this research, the temporal average will be calculated based on the last five seconds of the run duration.

4.3 Simulation Software

The software algorithms for the moving-bank MMAE/MMAC were developed and modified over the years in the previous thesis research [2,3,7,9,10,13,29,31,32,35]. The software will again be modified to meet the goals of this research. This software, which is resident on the Sun workstations, is broken into three separate computer program groups: (1) preprocessor, (2) processor, and (3) post-processor. Each of these program groups will be discussed individually in the following sections.

4.3.1 Preprocessor

Inherited Preprocessor. The previous preprocessor was separated into three separate routines which implemented a MATRIXx portion [18], a MATLAB portion [19], and a FORTRAN portion [5]. The MATRIXx code generated the truth and filter models for each point in the parameter space (this was the form of the models provided by the Phillips Laboratory). The MATRIXx truth and filter models were then reformatted to be compatible with MATLAB (the preferred software tool for this research for elemental filter or controller analysis and synthesis). The models were discretized, then the steady-state Kalman filter gains and the steady-state LQG regulator gains were calculated in MATLAB. Additionally, the calculation of the filter state covariance and residual covariance was performed in MATLAB. Finally, the MATLAB code stored the

information in a form compatible with the FORTRAN code for setting up files for actual Monte Carlo performance evaluations of the MMAE/MMAC algorithms. The remaining FORTRAN code was left with the task of reading the information from the MATLAB output and storing it in a form compatible with the processor code.

Current Preprocessor. There were a few minor changes made to the preprocessor programs. First, the MATRIXx and MATLAB routines both had to be modified to accommodate the increased number of positions in the parameter space. Also, the FORTRAN portion of the preprocessor was changed to allow the read and write functions to be done in an "unformatted" format, thereby significantly reducing the time required to read data. This also reduced the memory requirements for the filter/controllers corresponding to the 147 points in the parameter space which were taxing the limits of the memory allocated for this project.

4.3.2 Processor

Inherited Processor: The previous processor code (FORTRAN) simulated a moving-bank estimator/controller for the SPICE-4 structure with one uncertain parameter via Monte Carlo analysis. The FORTRAN code first read in information from data files which contained variables for the specific models used. The data from the preprocessor was read into the respective truth and filter bank matrices. Then the processor began the Monte Carlo loops by propagating the truth model and then the bank of filters. Measurements were taken from the truth model to be used in the update portion of the filter models. The respective filter probabilities were calculated and the decisions to move, contract, or expand the bank were made. The processor code included subroutines to implement the residual monitoring, probability monitoring, and the parameter position estimate monitoring methods for dictating the bank movement, as discussed in Section 2.4. Likewise, there were subroutines to implement MMAC, modified MMAC, MAP, and single changeable-gain control, as discussed in Section 2.5. The code was written to

accommodate bank expansion and contraction. The white Gaussian noise vectors used in both the dynamics driving noise and measurement noise generation were produced by a random number generator.

Current Processor: The FORTRAN portion of the processor code was changed to allow reading of data in an "unformatted" format, as explained above. The other major change to the processor involved increasing the parameter space and moving bank dimensions. In previous research, Gustafson [7] implemented a two-dimensional parameter space. This code was available for use, and was modified to add a third dimension. In most cases, this simply involved the addition of a third loop into the logic [4].

4.3.3 Post-Processor

Inherited Post-Processor: This FORTRAN code read data from the processor and generated the sample statistics (the mean and covariance) of the estimation position errors, the true LOS deviations with control applied, and the uncertain parameter estimation errors. The output generated by the post-processor was compatible with a separate plotting routine.

Current Post-Processor: A few minor formatting changes were made to the previous post-processor code and no major discrepancies were noted. The output generated was compatible with MATLAB plotting routines and was used for all the plots generated in the remaining chapters.

4.4 Analysis Plan

The main goal of this research is the effective estimation of the uncertain flexible bending modes, and control of the LOS deviations of the SPICE-4 structure, while allowing the parameters describing the flexible bending modes to vary independently. The flexible modes will be partitioned into three groups, the first group consisting of the first

three bending mode frequencies, the second group consisting of the next three frequencies, and the final group containing the remaining flexible bending mode frequencies. Each group of frequencies will have a separate scalar multiplier, thus resulting in a three-dimensional parameter space. This independent variation of the flexible bending modes is more representative of what happens in the "real world" than a single multiplier on *all* mode frequencies.

The 26-mode reduced-order SPICE-4 model was used by Schiller in developing his filters. Since the models used for this effort are the same, filter and controller tuning is not a concern [25,30]. The specific levels of dynamic driving noise and measurement noise are the same as for Schiller's research, as are the controller weighting matrices. For a complete development of these tuning parameters, see Schiller, Chapter 4 [31].

Prior to exploring the issues associated with the three-dimensional parameter space, the one-dimensional parameter space results obtained by Schiller [31] were reproduced in order to verify the system models and validate the software. The next two sections of this chapter will address the three-dimensional parameter space discretization and the parameter identification and control.

4.4.1 Sensitivity Analysis

As discussed in Section 3.6, the natural frequencies of the flexible bending modes of the structure form the basis of the uncertain parameter space. The actual discrete parameter points in the space are determined by performing a sensitivity analysis. This is accomplished by varying the scalar multiplier of the individual groups of the natural frequencies within the flexible bending modes in the truth model in one direction at a time until the closed loop system with the single filter/controller (based on the nominal value) in the loop becomes unstable. A discrete parameter value is declared just prior to this point (0.5 % rollback), where the difference from the nominal value is denoted as $\delta\omega_i$. Note that this parameterization method will result in the widest possible spread between each

point without any two points being so far apart as to produce unstable closed loop control.

Based on physical insights and the desire to keep the size of the problem reasonable, the full parameter space was chosen as 7x3x7, with the middle dimension limited to three discretization levels due to the insensitivity of the controller to the less dominant second group of bending modes. The sensitivity analysis was first performed in the direction of the second group of frequencies, with the other parameters left at nominal. This results in three discrete values for the second group of frequencies. Thus, the three-dimensional parameter discretization problem reduces to three separate planar discretizations.

Next, while keeping the second group of frequencies at nominal, a sensitivity analysis is again performed while varying the first and third groups of frequencies individually in both directions. The procedure is then repeated for the cases where both the first and third groups are allowed to vary together within the plane in each of four directions, thus creating a 3x3 "box" around the nominal value. To fill out the remainer of the parameter space in that plane, a new "nominal" is declared at the previous discretization points, and the sensitivity analysis is performed again in each of those directions, and the process is repeated until the entire space is filled. However, previous research [31] has shown that simply applying the same percentage change in each direction is an adequate way to fill out the remainder of the space; thus, this approach will be taken here.

Figure 4.2 illustrates this procedure in one dimension, where the size of the space and number of points is arbitrary and based on the results of the sensitivity analysis. In using this method, note that increasing the parameter results in a space determined where the filter is underestimating the natural frequencies. Similarly, decreasing the parameter from nominal results in a space determined where the filter is overestimating the natural

frequencies. For this problem, the parameter space was chosen as 7x3x7. For the first and third groups of bending modes, the space is discretized between the range of minus 4 to plus 8 percent from nominal with seven discrete values. The second group is limited to three levels, originally between minus 7.5 and plus 17 percent of nominal. Problems with this discretization level (to be discussed in Chapter 5) necessitated changes which limited the range of the second group also to minus 4 to plus 8 percent of nominal.

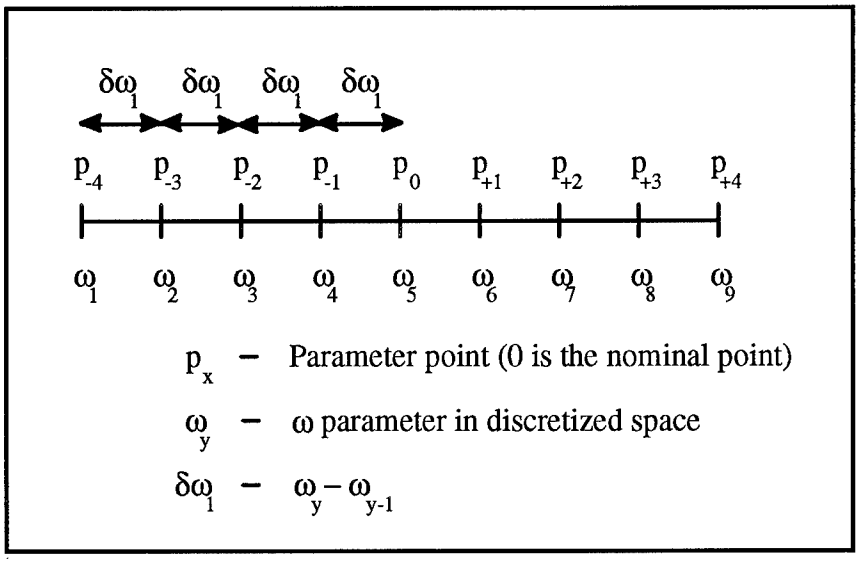


Figure 4-2. Parameter Discretization Procedure

4.4.2 Parameter Identification and Control

After the parameter space is fully discretized, simulations will be run to determine the effectiveness of the moving-bank MMAE/MMAC in identifying the true parameter and controlling the structure. Preliminary simulations indicated that the MMAE could indeed track a moving parameter with small state estimation errors while being run open-loop. Based on these results and on past research [31], all of the final simulations for this effort will be run closed-loop. The MMAE and the MMAC will be evaluated together

during closed-loop simulations. The moving bank will be a 3x3x3 cube in the 7x3x7 parameter space. In order to determine this capability, many simulations will be run in which the true parameter value will be positioned or moved in various fashions. These simulations will be similar to those run by Schiller [31]. First, in order to set a baseline of performance, the true parameter (the truth-model-based parameter value) will be set at the nominal value. Additionally, the moving bank of filters will be centered at this nominal value. Hence, the moving bank will be artificially informed initially of the true parameter value. Next, the true parameter value will be allowed to vary slowly over a period of time, which would simulate possible temperature variations or long term fatigue effects. The third type of simulation will center the filter bank on the true parameter initially, then allow the true parameter to make large discrete jumps to a new location. This condition simulates a situation that could occur during a structural failure. This type of offset scenario could also occur during an initial acquisition phase when the control system is first activated. These variations and jumps will be made in various directions throughout the parameter space.

4.5 Summary

This chapter has presented the performance analyses, software, and simulation plans involved with this research. The Monte Carlo analysis method was introduced, as well as the error vector formulation and the performance evaluation criteria. A brief discussion of the simulation software and the changes necessary to that software were provided. The chapter concluded with a discussion of the simulation analysis plan for evaluating the effectiveness of the MMAE/MMAC in estimating uncertain parameters and quelling unwanted vibrations.

Chapter 5 will present the results obtained from the parameter discretization efforts and the moving-bank MMAE/MMAC performance analysis.

V Results

5.1 Introduction

The primary purpose of this thesis is the design and performance evaluation of an adaptive control system that will quell the vibrations induced in the SPICE-4 structure. The undamped natural frequencies of the bending modes are assumed to be uncertain parameters present in the dynamics of the system. The frequencies are partitioned into three groups which are then allowed to vary independently of each other. It is assumed that a non-adaptive controller does not have enough inherent robustness to variations in the parameters, and thus cannot be implemented [17]. The concept of a multiple model adaptive estimator/controller is presented to provide the required quelling of the structure despite the uncertainties in the system.

The reduced-order design model used for this research was Schiller's modally-reduced, 26-mode model which contained 130 states [31]. This design model was chosen due to its proven effectiveness in parameter estimation and control of LOS deviations. Schiller found that the one dimensional MMAE/MMAC resulted in RMS LOS deviations of 0.682 micro-radians along the X-axis, and 1.012 micro-radians along the Y-axis for an artificially informed bank (the bank is centered on the true parameter value which is not allowed to vary) [31:5-6]. These results were verified in this research and will be discussed later. It should be noted that the numerical accuracy problems encountered by Schiller [31:5-9] in discretizing his parameter space with this model were not encountered in this effort.

This chapter will discuss first the discretization of the three-dimensional parameter space and the problems encountered in that process. Next a comparison of the straight-forward Bayesian vs. ME/I techniques will be provided. Determination of appropriate

threshold values will follow, and finally the performance of the MMAE/MMAC as it goes through many different parameter variation and jump scenarios.

5.2 Discretization of Parameter Space

Use of the sensitivity analysis method of discretizing the three-dimensional parameter space as described in Section 4.7 resulted in a severe problem. Recall that the first step in the discretization process was to vary only the second group of undamped natural frequencies. This group of frequencies has such a small effect on the overall control of the structure that it took a rather large deviation from nominal to cause the structure to go unstable. This discretization process led to a parameter variation of minus 7.5 to plus 17 percent of nominal with only three discretization levels for this specific parameter. The remainder of the parameter space was filled out in accordance with the procedure described in Section 4.7. Since the first and third groups exhibited stronger effects on control, their relative spacing in the parameter space was much smaller. Preliminary testing was then accomplished.

The moving-bank MMAE was found to perform well in open-loop simulations under variations in the first and third groups of frequencies. Expansion and contraction thresholds were determined by examining the likelihood quotients (recall Equation (2.32)) for all the filters during jumps across the parameter space in the direction of the first and third parameters. However, a problem arose whenever there was movement of the second group of frequencies. Because the discretization levels in this dimension were so severe, whenever there was movement in that dimension, the likelihood quotients for all the filters became large, indicating that none of the currently implemented filters had a viable model of vehicle bending characteristics. A simple move of one parameter position in this direction caused the likelihood quotients to be larger than a full six-position jump in the other two dimensions. Thus, whenever the parameter for the second group of frequencies

changed, the bank unnecessarily expanded regardless of any other movements. This type of expansion was totally unacceptable.

Therefore, a decision was made to rediscretize the parameter space while limiting the variation in the second group of frequencies to minus four to plus eight percent of nominal (same as the other dimensions). Again the discretization was limited to three levels. This resulted in a 7x3x7 three-dimensional parameter space with variations in all directions limited to minus four to plus eight percent of nominal. This discretization level resulted in markedly improved performance, as will be shown in detail in Section 5.6.

5.3 Bayesian Vs. ME/I Performance

Previous research has shown mixed results in comparing full (straightforward) Bayesian versus ME/I density computations in determining the parameter and state estimates. Gustafson [7] found that the ME/I method worked best on the SPICE-2 model, whereas Schiller [31] noted that the full Bayesian approach was more effective. Due to these conflicting results, these two methods, as well as removal of the beta-dominance term (as discussed in Section 2.3.2), were tested in this effort.

Preliminary testing of the full Bayesian method against a constant (and initially matched) or moving true parameter value demonstrated excellent results. The ME/I method was implemented by replacing the residual covariance matrix, $\mathbf{A}_k$, in the probability density function for the current measurement with an identity matrix. Recall Equations (2.26) and (2.31). This replacement caused a problem in the density calculation due to the very small absolute size of the residuals. The quadratic term $\left[\mathbf{r}_k^{\mathrm{T}}\mathbf{I}\mathbf{r}_k\right]$ in Equation (2.31)) was so small that the resulting exponential approached "1" for each of the individual filters in the bank (e.g., $e^{-.00001} = 0.99999$). This resulted in the inability to differentiate between correct and incorrect filters since all filters had basically the same scaled probability. Therefore, the parameter estimate never changed.

In order to alleviate this problem, the identity matrix in the density calculations is scaled in each filter by the same scalar value. This allows for a greater variation in the density calculation by providing exponentials that are more usable. To determine the best performance with this method, the scaling factor was allowed to vary. A scaling factor of 500,000 was determined to be best by trial and error. However, while the probabilities were now able to change, the parameter estimate with the ME/I method was not as effective as in the pure Bayesian method that retained $\mathbf{A}_k$ in Equation (2.26). The best ME/I method still exhibited an extremely sluggish response time to true parameter changes, and also had a very erratic response during tracking.

The poor results of the ME/I method are most likely due to the lack of proper scaling of the residuals. The two types of measurements, position and velocity, have an order of magnitude discrepancy in accuracy. Thus the resulting residuals have varying orders of magnitude. The position measurements are relatively more accurate, thus have smaller residual values. These small residual values can be obscured by the higher magnitude velocity residuals in the equally scaled quadratic. Having different scale factors for different residual types might have some success, but having too many scale factors is counteractive to the determination of the smallest absolute residuals. This approach has been attempted previously [7], but it resulted in limited sucess.

5.4 Bayesian MMAE/MMAC Performance

This section presents the results obtained from the MMAE/MMAC design method. Several techniques discussed in Section 2.3 through Section 2.6 were used to move the bank and implement control to the structure. Of the four methods for moving the filter bank, only three methods were implemented. Comparisons of residual monitoring, probability monitoring, and parameter position monitoring were carried out through the MMAE/MMAC analysis. Parameter position and velocity estimate monitoring was not performed due to poor results of earlier applications of this method [9,26].

In implementing the MMAC portion of the design, several techniques discussed in Section 2.6 were utilized. These control methods include: MMAC, modified MMAC, MAP, and modified single changeable gain methods. The MMAC method blends the control inputs from each of the active filter/controllers based on their probability weights. Modified MMAC also provides a blending of control inputs from each of the filters; however, it institutes a lower bound on the probabilities which precludes the blending of filter/controllers whose probability falls below this threshold. The MAP method declares the filter/controller with the highest probability as the one to provide all of the control input. The modified single changeable gain method uses the parameter estimate from the MMAE, interpolates the controller gain from the four closest filter/controllers based on that parameter estimate, and cascades that with the state estimate to form the control inputs. Fixed gain and single changeable gain methods were not implemented due to the excellent results achieved with the above methods.

Simulations conducted for this research consisted of a comparison of each combination of the bank movement and control methods, carried out over a variety of different parameter movement and jump scenarios, as discussed in Section 4.4.5. Initial results indicated that the open-loop MMAE could track a varying true parameter. Thus, for the final runs, both the MMAE and MMAC will be evaluated during closed-loop simulations. Each simulation consisted of a ten-run Monte Carlo analysis with a ten-second simulation time (except where noted). The controller is turned on half a second into the simulation for all runs to allow the estimator time to develop an accurate state and parameter estimate.

5.5 Threshold Determination

Prior to any simulations being run, a variety of thresholds needed to be determined. These included probability limits, bank movement thresholds, expansion thresholds and contraction thresholds. Recall from Section 2.3.2 that a lower bound is necessary on the

probability weights to prevent the lockout condition from occurring. Due to the large number of filters in this effort, the resulting lower probability limit is very small. There is a tradeoff in setting this limit. Too high a limit will result in erroneous weights given to incorrect filters/controllers. Too low a limit results in the inability of the probability to change in a timely manner in response to real parameter changes, due to the iterative nature of the calculation in Equation (2.25). By trial and error, the best value for this lower probability limit was determined to be 0.001.

The probability limit for inclusion in the modified MMAC calculation was chosen to be 0.1. Any filter/controller whose probability is below this threshold is not included in the control calculations. This precludes erroneous control from being input to the system. This threshold must be higher than the artificial lower bound on the probabilities $p_k(t_i)$. The value for this threshold was determined by trial and error. The threshold value was varied to determine which value produced the smallest line-of-sight errors during a simulation in which the estimator tracked the true parameter. A value of 0.1 resulted in the smallest rms errors.

Bank movement thresholds must be determined for each of the three move types. Since they are each based on different hypotheses for movement, they will all have different thresholds. In each case there is a tradeoff between trackability (variation of the parameter estimate around the true value in an unchanging true parameter condition), and response time (how quickly the parameter estimate can adapt to a true parameter change). Setting the move threshold too low would result in unnecessary move decisions, whereas setting the threshold too high would result in slow or no response to actual parameter changes.

For the parameter position monitoring technique, the move threshold is based on the difference between the bank center and the parameter position estimate. Since it is only necessary to move the bank when the parameter estimate is more than half way to a

different parameter position, the move threshold for this method is set to 0.5. If this threshold is exceeded, the bank is moved in the direction of the new parameter estimate. In the probability monitoring technique, the move threshold basis is the magnitude of the conditional probabilities. If the conditional probability of the filter with the highest probabiltiy value exceeds the threshold, then a decision is made to move the bank in the direction of this filter. Due to the large number of filters in this effort, the probability "move" threshold had to be significantly reduced from previous efforts. By trial and error, the probability monitoring "move" threshold was determined to be 0.1. This allowed quick bank movement once the probabilities started to floe between the filters. For the residual monitoring technique, the move logic is based on the value of the likelihood quotient, L, for each filter (recall Section 2.4.2). If the threshold value is exceeded by all the filters in the bank, then a decision is made to move the bank in the direction of the filter with the smallest L. Again by trial and error, the move threshold for this technique was determined to be 11.0.

The bank expansion threshold was determined based on the likelihood quotients as well. The expansion threshold must be set higher than the residual monitoring "move" threshold. If the likelihood quotient for all the filters exceeds the expansion threshold, the bank is expanded so that the filters are relocated at their widest possible locations in the parameter space. The size of the likelihood quotient indicates that the true parameter position is not in the vicinity of the current filter bank. The threshold level was determined by trial and error to be 50. A tradeoff had to be made between setting the threshold too high which did not allow proper expansion of the bank, and setting the threshold too low which resulted in unnecessary expansions. Due to the small relative size of the parameter space, the final value of 50 allowed expansion only when the true parameter was at least four (fine discretization level) parameter positions from the bank center.

After the bank has expanded, contraction takes place when the variance of the parameter position estimate falls below a set threshold. Since the bank expands in two of the three parameter space directions, two contraction thresholds are used, one for each of the expansion directions of the parameter space. This allowed for the possibility of rectangular block as well as cubic filter banks. Both contraction thresholds were set to 1.5, but since the third group of frequencies was generally easier to estimate, the third direction usually contracted first. In both parameter directions, the contraction thresholds were both set very high, which resulted in quick contractions, typically within 5 sample periods after an expansion had occurred.

The expansion/contraction logic was designed to have the bank expand to its widest positions, given a large change in the true parameter position. In this way the true parameter position would be guaranteed to be encompassed within the bank and could be estimated. The bank would then contract around that value. It was anticipated that the expanded bank could estimate the true parameter position somewhere in between the discrete parameter points of the expanded bank. However, as implemented here, it could not. In an expanded state, all probability flowed into the filter that was closest to the true position, and blending of the estimates was not achieved. Thus, the contraction logic consistently placed the center filter at one of the expanded filter bank's discrete parameter locations. Schiller [31] noticed the same effect in his research. Because of the greater relative size of his parameter space, different levels of contraction (coarse, intermediate, and fine) were necessary. Due to the smaller size of the parameter space in this effort, the contracted bank was never more than one position away from the true value, and the bank move logic was then implemented.

The software was reviewed, but no errors could be found that would cause this situation to occur. Thus, the contraction thresholds were set high, which allowed the bank to contract quickly, then move according to the move logic. The contraction logic usually

made the correct decision to center the bank on the filter closest to the true parameter. Occasionally, a wrong decision was made, but the bank invariably either moved to the correct location, or expanded again until a correct decision was made.

5.6 Results

As stated previously, each combination of move method and control method was tested against a variety of true parameter move and jump scenarios. A move scenario occurs when the true parameter value is allowed to vary by only one parameter position at a time, whereas a jump scenario occurs when the true parameter moves six parameter positions at any one time across the parameter space. First, a filter bank that was initially artificially informed of the parameter value was matched to an unchanging true parameter. Next, the individual parameters (the first group of frequencies is designated a_1, the second group a_2, and the third group a_3) were allowed to vary individually. Then two parameters were allowed to vary simultaneously. Finally, two random runs were conducted which show the ability of the bank to track while all three parameters are moving simultaneously, as well as to demonstrate the ability of the bank to estimate a true parameter that is at the edge of the parameter space. Jump simulations were conducted in a similar manner, first with individual parameter jumps, then two parameters jumping simultaneously, and finally with all three parameters jumping together. Results from all these simulations are plotted in the Appendix. In evaluating the different controller types, it was noted that, with the modified single changeable gain method the estimator had a tendency to lose lock on the true parameter value. This only happened when both a_2 and a_3 were simultaneously decreasing. The code was reviewed, but no errors could be found that would cause this to happen. However, it may be due to the specific gain interpolation method used in the algorithm. The algorithm first estimated the a_2 position, and then used a linear combination of the controller gains of the four closest filters/controllers *in that plane* to determine the control input. Thus a poor estimate in the a_2 direction would

cause inappropriate control in the structure, which may have caused the bank to lose lock on the parameter value. However, in some cases, "inappropriate" control might make a parameter more identifiable. For this reason, the modified single changeable gain method was determined to be unacceptable. Perhaps a better method would have been to combine the closest eight filters in a 2x2x2 cube around the current estimate to provide the control inputs. This would reduce the susceptibility to noise in the a_2 direction.

For the other three control methods, the LOS deviations did not depend significantly on the method being applied. In fact, in most cases the actual LOS errors were the same to the third significant figure for any control method. The modified MMAC method did exhibit a slight improvement over the other two methods. This is to be expected, since the modified MMAC is a combination of the Bayesian MMAC and the MAP MMAC methods. During a time of changing parameters, the modified MMAC will allow a blending of control inputs similar to the Bayesian MMAC, but during tracking phase, all control will come from one controller since the other controller's probabilities will most likely fall below the threshold for inclusion in the control input. Therefore, the plots shown in the Appendix are only those for the modified MMAC results, although both the Bayesian MMAC and MAP methods also exhibited excellent (virtually identical) results.

In evaluating the different move methods, the probability monitoring and parameter position monitoring methods exhibit a clear advantage over the residual monitoring technique. In residual monitoring the move decision is based only on the latest residual value, not the entire past history. Thus, this method is plagued by single large noise samples causing unnecessary move decisions. In the probability and parameter position monitoring methods, the move logic is based not only on the current measurements, but also on the past history of measurements. Thus, single large residual samples do not have such an adverse effect, which results in better overall performance.

Schiller [31] found that the parameter position monitoring method was more erratic than the probability monitoring method. However, an error was discovered in his code which caused the bank to move twice as far as intended, thus his implementation of this method exhibited an oscillatory movement about the true parameter value. Once this code was corrected, the probability monitoring method and the parameter position estimate method were seen to be almost identical.

For each of the simulations run, the results are shown in the Appendix. The plots are grouped into sets of five for each simulation type. The first three plots in each group exhibit the parameter identification results with each of the three move types with modified MMAC control. The fourth plot in each group represents the MMAE estimation errors, while the fifth plot shows the MMAC rms line-of-sight errors. These last two sets of error plots result from the parameter position estimate move method and the modified MMAC control method; however, all move methods exhibited virtually identical results.

5.6.1 Initially Matched Filter

Initially matched filter simulations were run such that the bank was initially centered on an unmoving nominal true parameter. Figures A-1 through A-3 exhibit the actual parameter estimation plots for each of the three move methods. It is noted that, for the parameter position and probability methods, the estimate identically matches the true value after a short transient period. Figure A-1 shows the mean $\pm 1\sigma$ values, while Figures A-2 and A-3 exhibit just the mean value of the parameter estimates. In all cases, the $\pm 1\sigma$ values are very small. Thus, except for a few sample cases, it was decided not to include $\pm 1\sigma$ plots because the 1σ values were very small for all but the short transient period following a true parameter move, and the addition of the extra lines would cause undue clutter to the plots.

"Move" Logic:	Residual	Probability	Parameter Position
Initially Matched Filter	X = 0.707 μrad Y = 1.039 μrad	X = 0.682 μrad Y = 1.012 μrad	X = 0.682 μrad Y = 1.012 μrad

Table 5-1. Initially Matched Filter Temporally Averaged RMS LOS Errors

Figure A-4 presents the X and Y axis estimation errors. Figure A-5 shows representative X and Y axis LOS deviations, i.e., performance of the controlled variables of interest. Notice the effect of turning the controller on after an initial half second without MMAC control. Temporal averages, taken over the last 8 seconds of the simulation for the three different move types, are presented in Table 5-1. These results are consistent with Schiller's results [31].

5.6.2 Parameter Variations (Moves)

As the true parameters were allowed to move slowly throughout the parameter space, each move method accomplished an adequate job of estimating the parameter and controlling the structure. Again, the results are partitioned into groups of five for each simulation. The first three plots show the parameter identification performance for each move method, followed by the estimation and line-of-sight errors for the parameter position estimate method. As expected, the parameter position estimate method and the probability monitoring method were slightly better than residual monitoring. Again, the estimation errors and LOS deviations are nearly identical for all cases. Figures A-6 through A-30 depict the results of the simulations in which one parameter is moving at a time. Notice that the a_2 and the a_3 estimates have no trouble in tracking the true parameter movements. For the a_2 direction, the extreme levels of discretization makes it easy to determine which residuals are good or bad, thus making it easy to track this parameter. For the a_3 direction, the ease of tracking is due to the multitude of

information contained in the large group of modes at various frequencies. The estimator had a harder time in the a_1 direction due to the small number of modal frequencies in that group and the tight level of discretization in that direction. Also note that the estimator had a harder time tracking when the a_1 parameter was decreasing than when it was increasing. See Figures A-11 through A-13. Although the lowest frequency modes effect the most control on the structure, this relatively poor tracking performance does not have a debilitating effect on the stability and control of the structure. When the estimator *overestimates* the true parameter value, the controller can continue to quell bending oscillations in the structure rather well. However, when the estimator *underestimates* the true parameter value, the LOS deviations exhibit degraded control of the structure. This is consistent with the performance observed in discretizing the parameter space and developing the LQG controller for each discrete point value. The degradation for such a *nonadaptive* controller that underestimates the undamped natural frequencies is much worse than if it overestimates them. For the case when a_1 is decreasing the simulation time between true parameter jumps was doubled to ensure that the estimate did converge to the true value, and it did.

Figures A-31 through A-50 show simulations when a_1 and a_3 were allowed to vary simultaneously in a variety of different directions throughout the parameter space. Figures A-51 through A-60 show a_1 and a_3 moving when a_2 was not at its nominal value. Figures A-61 through A-70 exhibit the results of the 20-second "random" runs in which the three parameters were all allowed to move simultaneously. In these runs the true parameter value was also allowed to move to the edge of the parameter space. Notice that the a_1 estimates tended to lose track at the very end of the first random move simulation. As seen in Figure A-65, this does not have a destabilizing effect on the structure due to the fact that the estimator is overestimating the true parameter. In the second random move simulation, the X and Y LOS deviations are significantly degraded,

"Move" Logic:	Residual	Probability	Parameter Position
a_1 moving up	X = 0.749 µrad Y = 1.100 µrad	X = 0.736 µrad Y = 1.071 µrad	X = 0.736 µrad Y = 1.071 µrad
a_1 moving down	X = 0.720 µrad Y = 1.052 µrad	X = 0.716 µrad Y = 1.041 µrad	X = 0.716 µrad Y = 1.041 µrad
a_2 moving	X = 0.723 µrad Y = 1.060 µrad	X = 0.698 µrad Y = 1.019 µrad	X = 0.695 µrad Y = 1.017 µrad
a_3 moving up	X = 0.701 µrad Y = 1.008 µrad	X = 0.668 µrad Y = 0.980 µrad	X = 0.668 µrad Y = 0.980 µrad
a_3 moving down	X = 0.719 µrad Y = 1.079 µrad	X = 0.698 µrad Y = 1.055 µrad	X = 0.698 µrad Y = 1.053 µrad
a_1 and a_3 moving up	X = 0.707 µrad Y = 1.033 µrad	X = 0.710 µrad Y = 1.026 µrad	X = 0.710 µrad Y = 1.026 µrad
a_1 and a_3 moving down	X = 0.723 µrad Y = 1.084 µrad	X = 0.707 µrad Y = 1.060 µrad	X = 0.704 µrad Y = 1.058 µrad
a_1 moving up/a_3 moving down	X = 0.756 µrad Y = 1.112 µrad	X = 0.744 µrad Y = 1.102 µrad	X = 0.745 µrad Y = 1.103 µrad
a_1 moving down/a_3 moving up	X = 0.700 µrad Y = 1.020 µrad	X = 0.674 µrad Y = 0.985 µrad	X = 0.671 µrad Y = 0.980 µrad
$a_2 = 1$, a_1 and a_3 moving up	X = 0.700 µrad Y = 1.013 µrad	X = 0.698 µrad Y = 1.017 µrad	X = 0.698 µrad Y = 1.016 µrad
$a_2 = 3$, a_1 and a_3 moving up	X = 0.697 µrad Y = 1.009 µrad	X = 0.698 µrad Y = 1.008 µrad	X = 0.700 µrad Y = 1.010 µrad
Random Run #1	X = 0.688 µrad Y = 1.024 µrad	X = 0.673 µrad Y = 1.007 µrad	X = 0.671 µrad Y = 1.005 µrad
Random Run #2	X = 1.962 µrad Y = 2.653 µrad	X = 0.903 µrad Y = 1.234 µrad	X = 0.795 µrad Y = 1.179 µrad

Table 5-2. Parameter Variation Temporally Averaged RMS LOS Errors

as seen from Table 5-2. This occurs due to the fact that the estimator is underestimating the a_1 parameter for significant amounts of time. As stated previously, underestimation of the parameter values tends to have a serious degrading effect on control. Table 5-2 summarizes the X and Y axis rms LOS errors, temporally averaged over the last 8 seconds of the simulation for each of these simulations, indicating the ability of the MMAC to control the structure. The X axis LOS errors always fall well within the specified one micro-radian, whereas the Y axis errors do exceed the specification by a few percentage points during parameter movement. The parameter position and probability monitoring move methods exhibit somewhat smaller errors than the residual monitoring method.

5.6.3 Parameter Jumps

As the true parameters were allowed to jump from one edge of the parameter space to the other, the moving bank MMAE/MMAC provided adequate tracking of the parameters and controlling of the structure. Figures A-71 through A-85 show parameter identification plots as well as estimation and LOS errors when one parameter at a time is allowed to jump. Notice in Figures A-71 through A-73 that, when parameter a_1 jumped, the bank did not expand and contract, but simply stayed at the finest discretization level (smallest bank size) and slid across the parameter space to the true value. As described in Section 5.1, movements in the a_2 direction force a lower limit onto the expansion threshold to preclude bank expansion during true parameter movement in that direction. Therefore, the expansion threshold could not be lowered so that the bank would expand in this case. However, this slow movement across the parameter space does not seriously inhibit the controller, as seen from Figure A-75 and the values in Table 5-3. Table 5-3 presents the X and Y axis rms LOS errors temporally averaged over the last 8 seconds of each of the jump simulations. In every other case of parameter jumps, the bank did expand and contract (except for the case of a jump in a_2 where expansion or movement is not warranted) as desired.

"Move" Logic:	Residual	Probability	Parameter Position
a_1 jumps	X = 0.790 μrad Y = 1.077 μrad	X = 0.732 μrad Y = 1.059 μrad	X = 0.732 μrad Y = 1.059 μrad
a_2 jumps	X = 0.699 μrad Y = 1.028 μrad	X = 0.694 μrad Y = 1.025 μrad	X = 0.692 μrad Y = 1.025 μrad
a_3 jumps	X = 0.718 μrad Y = 1.094 μrad	X = 0.712 μrad Y = 1.092 μrad	X = 0.711 μrad Y = 1.089 μrad
a_1 and a_3 jumps (same direction)	X = 0.733 μrad Y = 1.116 μrad	X = 0.729 μrad Y = 1.108 μrad	X = 0.729 μrad Y = 1.108 μrad
a_1 and a_3 jumps (opposite direction)	X = 1.975 μrad Y = 4.010 μrad	X = 2.173 μrad Y = 3.924 μrad	X = 2.037 μrad Y = 2.991 μrad
All three jump a_1 and a_3 opposite	X = 0.941 μrad Y = 1.447 μrad	X = 0.916 μrad Y = 1.336 μrad	X = 0.928 μrad Y = 1.342 μrad
All three jump a_1 and a_3 same	X = 0.729 μrad Y = 1.115 μrad	X = 0.724 μrad Y = 1.113 μrad	X = 0.720 μrad Y = 1.112 μrad

Table 5-3. Parameter Jumps Temporally Averaged RMS LOS Errors

For a decreasing jump in the a_3 direction (see Figures A-81 through A-83), the estimate does not converge back to the true value. This is not seen as a terrible problem, as adequate control is still input to the structure, as seen from Figure A-85. Again, this is due to the fact that the controller does a better job when the estimator overestimates the true parameter value than when it underestimates the value. Figure A-81 also shows an example of the the mean $\pm 1\sigma$ plots. Notice that, even during a time of bank expansion, the $\pm 1\sigma$ values are very small, which indicates that the results are very consistent from run to run. Although this consistent bias is of concern, the estimator is overestimating,

versus underestimating, the parameter, and the LOS performance degradation is not significant.

Figures A-86 through A-95 show simulations when a_1 and a_3 jump simultaneously. When the parameters jump in the same direction, no problems are detected. However, when the jumps are in opposite directions, the estimator exhibits some difficulty in parameter estimation. Although the MMAE eventually does converge to the true values, the relatively long period of underestimation of the more predominant a_1 modes causes the beginnings of instability to be evident in the X and Y LOS deviation plots (see Figures A-94 and A-95). The large LOS errors do settle out as the estimator converges on the true parameter value.

Finally, Figures A-96 through A-105 show the ability of the algorithm to track three simultaneous parameter jumps. Notice that, when a_1 and a_3 jump in opposite directions (Figures A-96 through A-98), the MMAE does not have the previously seen estimation problem when a_2 is moving as well. Also, notice that, when the three parameters all jump in the same direction (Figures A-101 through A-103), the a_1 estimate takes a long time to track to a positive jump. This does not seem to affect the control of the structure as indicated by Figure A-105 and the values in Table 5-3. This is a strange observation since underestimation of the true parameter value usually has an adverse effect on control. This does not appear to be the case for this instance.

In each of the jump simulations presented in Table 5-3, the X axis rms LOS errors are within the specified one micro-radian. The Y axis rms errors are only a few percentage points above the specification. This difference between axes is due to the geometry of the X and Y axes relative to the location of the three legs of the tripod structure connecting the two assemblies in Figure 1-3. The only exception to this is when a_1 and a_3 jump in opposite directions. Only then does the MMAE/MMAC exhibit any

kind of destabilizing control. As previously noted, the parameter position and probability move methods exhibit slightly lower errors than the residual monitoring technique.

5.7 Summary

This chapter presented the results of this research. The chapter began with a discussion of the parameter space discretization. The extreme levels of discretization in the second group of frequencies necessitated altering the parameter discretization method to limit the parameter bounds to minus four to plus eight percent of nominal in all directions. In the MMAE formulation, the Bayesian form of the probability density function calculation given by Equation (2.26) performed much better than the ME/I formulation given by Equation (2.31).

Choosing the appropriate threshold values was shown to be an important design step. Setting thresholds too high inhibited bank movement or expansion, while setting thresholds too low resulted in unnecessary expansion or movement. Thresholds were determined by trial and error, and tradeoffs were necessary to obtain the best performance.

All three bank movement techniques dispalyed excellent parameter tracking characteristics. Residual monitoring was slightly more erratic than the other two, as was expected. The estimator had no trouble in tracking a_2 and a_3, but exhibited some sluggishness in the a_1 estimate (associated with the lowest frequency bending modes). This is due to the small number of modes present in this grouping, and the small level of discretization between points in this direction of the parameter space.

Three of the four control techniques proved to be acceptable. The modified single changeable gain method was abandoned after the estimates lost lock in a few isolated cases. Of the three remaining methods, the modified MMAC was slightly better than the Bayesian MMAC and MAP MMAC methods at reducing the LOS deviations. The controller did perform better when the estimator overestimated, versus underestimated,

the true parameter value. When the estimator underestimated the parameter, some substantial degradation of control was exhibited.

VI Conclusions and Recommendations

6.1 Introduction

The focus of this research has been to apply moving-bank multiple model adaptive estimation and control (MMAE/MMAC) algorithms to quell undesirable vibrations in the SPICE-4 space structure. The necessity for adaptive control was based on concerns that unadaptive controller designs do not demonstrate enough inherent robustness to be effective in the presence of uncertain parameters in the system model [16]. In this application, there were three uncertain parameters. The undamped natural frequencies of the flexible body modes of the physical structure were split into three groupings. The first grouping contained the three lowest frequency modes, the second grouping contained the next three lowest frequency modes, and the third grouping contained the remaining flexible body modes. The uncertain parameters were the three scalar multipliers for the frequencies of each of these groups. Thus the groups of frequencies were allowed to vary independently of each other. The SPICE-4 truth and reduced-order filter models were developed by Schiller in previous research [31]. It was decided to use the 26-mode, modally reduced filter model based on its success in Schiller's research. The parameter space was discretized in three dimensions, and the inherited simulation software was modified to accommodate the three dimensional parameter space. The ability of a moving bank of filter/controllers to estimate the uncertain parameter accurately and apply control effectively was examined as the true parameter wandered and jumped throughout the parameter space.

6.2 Conclusions

The moving-bank MMAE/ MMAC design method is extremely effective in quelling unwanted vibrations in the SPICE-4 structure even during periods of large parameter variations. Discretization of the parameter space was determined to be a key

design procedure in the three dimensional space. Since the different groupings have differing effects on the control of the structure, applying the sensitivity analysis method resulted in large variations of the discretization levels, especially in the direction of the less sensitive second group of undamped natural frequencies. Therefore, because the residuals were so large in all elemental filters with an incorrectly assumed value for the parameter in that direction, the likelihood quotients would become distorted any time there was movement of the true parameter in that direction. This would cause undue expansion of the filter bank and unnecessary degradation of the parameter estimate. When the parameter space was rediscretized to limit the amount of variation in the direction of the second group, this unnecessary expansion no longer occurred, and estimation and control perfromance improved markedly. This seems to indicate that the best way to discterize the three dimensional space is to allow only the same percentage change between discrete parameter points in *all* directions.

Of the two formulations in the residual probability density function used for computing the hypothesis probability weights, the full-scale Bayesian form resulted in much better parameter position estimate than the Maximum Entropy with Identity residual covariance method (ME/I). Although all of the three move logic techniques investigated were very effective at parameter identification, the residual monitoring method was not quite as good due to the tendency to be impacted by single large measurement noise samples. The parameter position estimate and probability monitoring techniques were seen to be virtually identical. In any case, the determination of the bank movement and expansion thresholds was of utmost importance. Choosing thresholds too low resulted in unnecessary bank movement or expansion, and choosing thresholds too high caused the bank not to move or expand when it was warranted.

Of the four control methods implemented, the modified single changeable gain method exhibited some difficulties in maintaining good parameter estimates in a few

isolated instances. The other three methods resulted in nearly indistinguishable performance characteristics, with the modified MMAC method providing slightly better control of the line-of-sight errors. During times of parameter tracking, the one micro-radian specification on the LOS deviations was exceeded only in the Y-axis direction, and only by about one percent. (The X and Y characteristics differ due to the tripod geometery of the legs of the SPICE-4 structure.) The X-axis LOS errors were well within the specification. During times of parameter moves and jumps, the X-axis LOS errors were still within specification except in the case where a_1 and a_3 make jumps in opposite directions, but the Y-axis LOS errors fell outside the specified value in most instances, getting as large as 3 micro-radians in the case of a_1 and a_3 jumping in opposite directions. Once the short transition period after a parameter move was expired, the Y-axis LOS error once again resided on the specification. An important point to note is that the controller is more effective at quelling the LOS errors when the estimator overestimates, versus underestimates, the true parameter value.

These results indicate that the MMAE/MMAC algorithms will provide highly effective control for the SPICE-4 structure, even with large parameter variations. The following section presents areas of future research.

6.3 Recommendations

This research demonstrated the effectiveness of the three dimensional moving-bank MMAE/MMAC algorithms as applied to the SPICE-4 structure. Recommendations for future research are:

1. Explore the effects of dither inputs to the SPICE-4 system model to enhance parameter identification. Investigate both open-loop dithering to enhance the *initial* parameter identification at onset of control being applied, and possibly closed-loop dither applied continuously or intermittently to enhance parameter identifiability and control performance while in closed-loop control operation.

2. Investigate possible alternatives in grouping the different frequencies such that all groupings have more equal effects on the control of the structure.

3. Apply the modal-cost order reduction technique to determine which groupings of frequencies will have the most impact on control, and compare the results using the modal order reduction employed in this research.

4. Try a parameter space discretization in which the moving bank will be allowed to move in all three dimensions rather than just two.

5. Investigate further the reasons why an expanded bank sometimes cannot provide a correct estimate when the true parameter value is between the expanded bank's filter locations.

6. Utilize the latest version of the SPICE models if they are available from the Phillips Laboratory, or possibly implement the moving-bank MMAE/MMAC on a new and different structure.

7. Investigate the possibility of implementing a simple windowing procedure for the residual monitoring technique in which the most recent N measurement samples are utilized in the move logic vice the single most recent measurement sample. This may decrease parameter estimation variation without a large increase in the response time.

Appendix: MMAE/MMAC Performance Results

This appendix presents all the plot results for the MMAE/MMAC analysis in Section 5.6. The plots in this appendix represent a very good representative sample of all the actual simulations conducted for this section. For each type of parameter move or jump, the following plots are separated into groups of five. For each simulation run, parameter identification plots are shown for each of the three move methods with the modified MMAC control implemented (one of four actually investigated, but the one providing the best performance). Next, X and Y axis estimation errors are shown, followed by the X and Y axis LOS errors. The error plots were virtually identical for all three move methods, thus only the parameter position estimate method is shown, again with the modified MMAC control.

Although there is no legend associated with each of the parameter identification plots, it should be fairly evident that the true parameter is indicated by the straight or piece-wise constant lines and the parameter estimate (mean value obtained from a ten-run Monte Carlo analysis) is indicated by the wavering lines. Except for a few sample cases, it was decided not to include mean $\pm$ 1σ plots because the 1σ values were very small for all but the short transient period following a true parameter move, and the addition of the extra lines would cause undue clutter to the plots. In the cases where the 1σ plots are shown, the $\pm$1σ values are indicated by dashed lines. Likewise, on the estimation and LOS error plots, solid lines indicate the mean error from the ten-run Monte Carlo analysis, while the $\pm$1σ values are indicated with dashed lines.

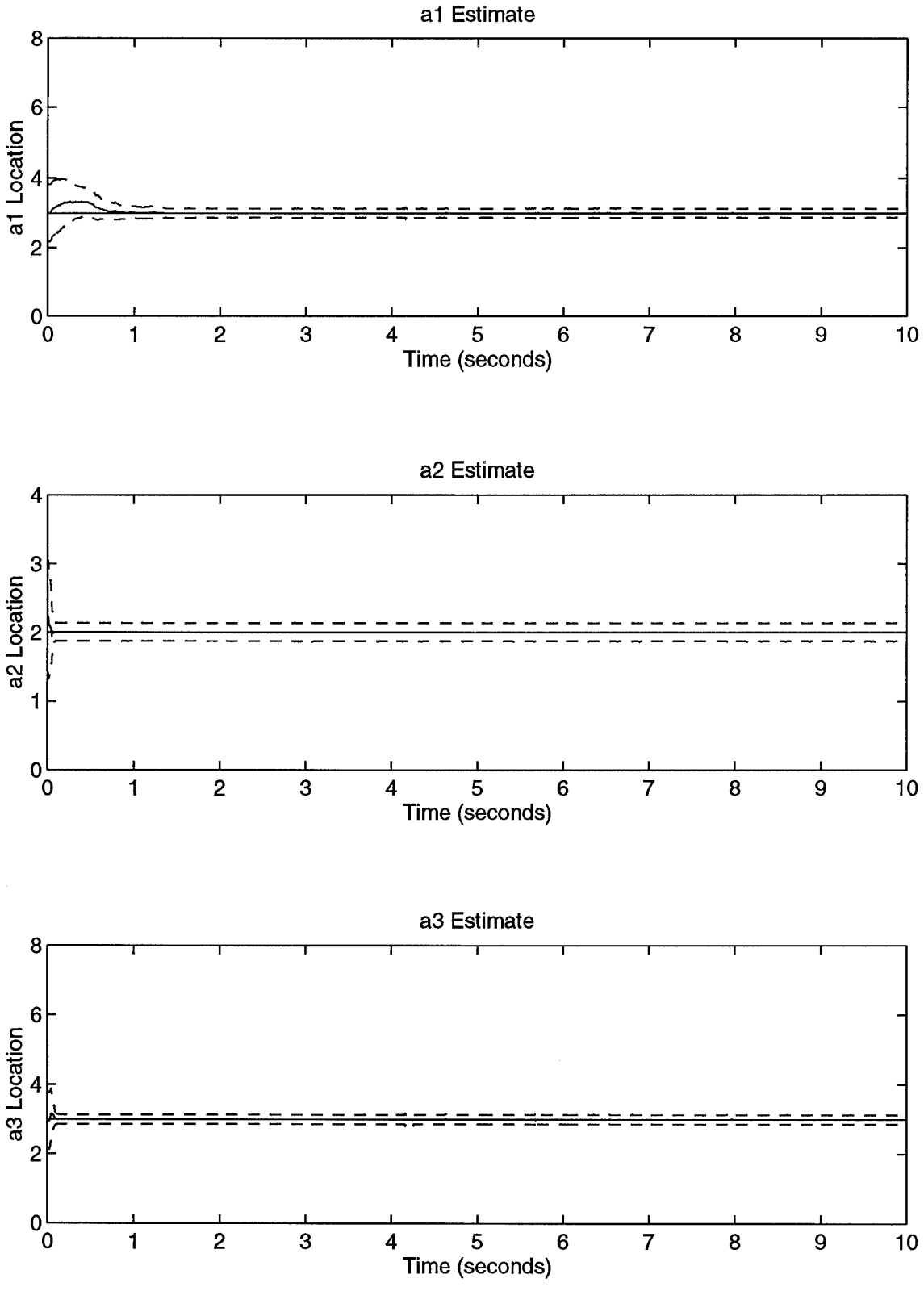


Figure A-1. Initially Matched Filter Using Parameter Position Monitoring

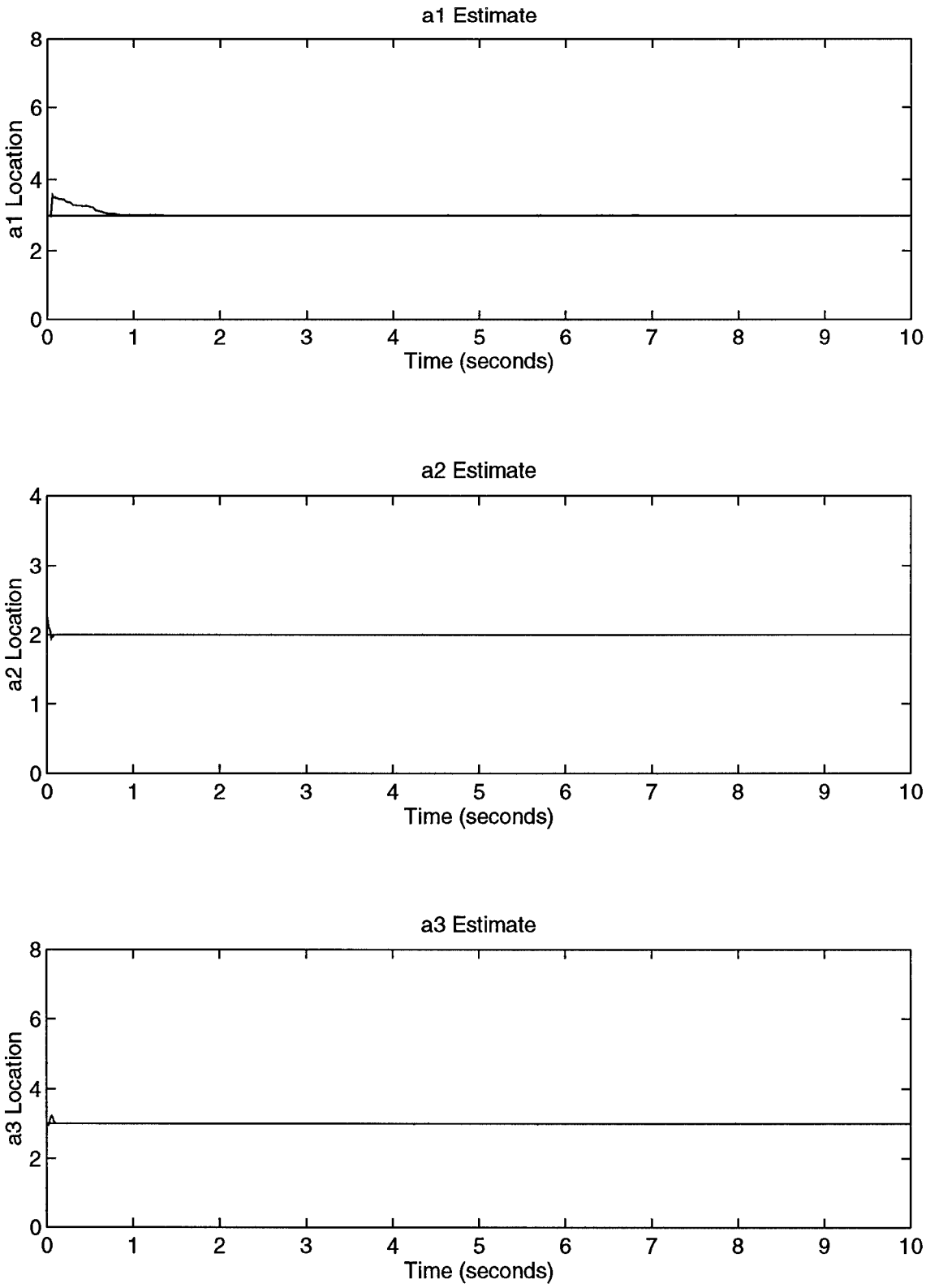


Figure A-2. Initially Matched Filter Using Probability Monitoring

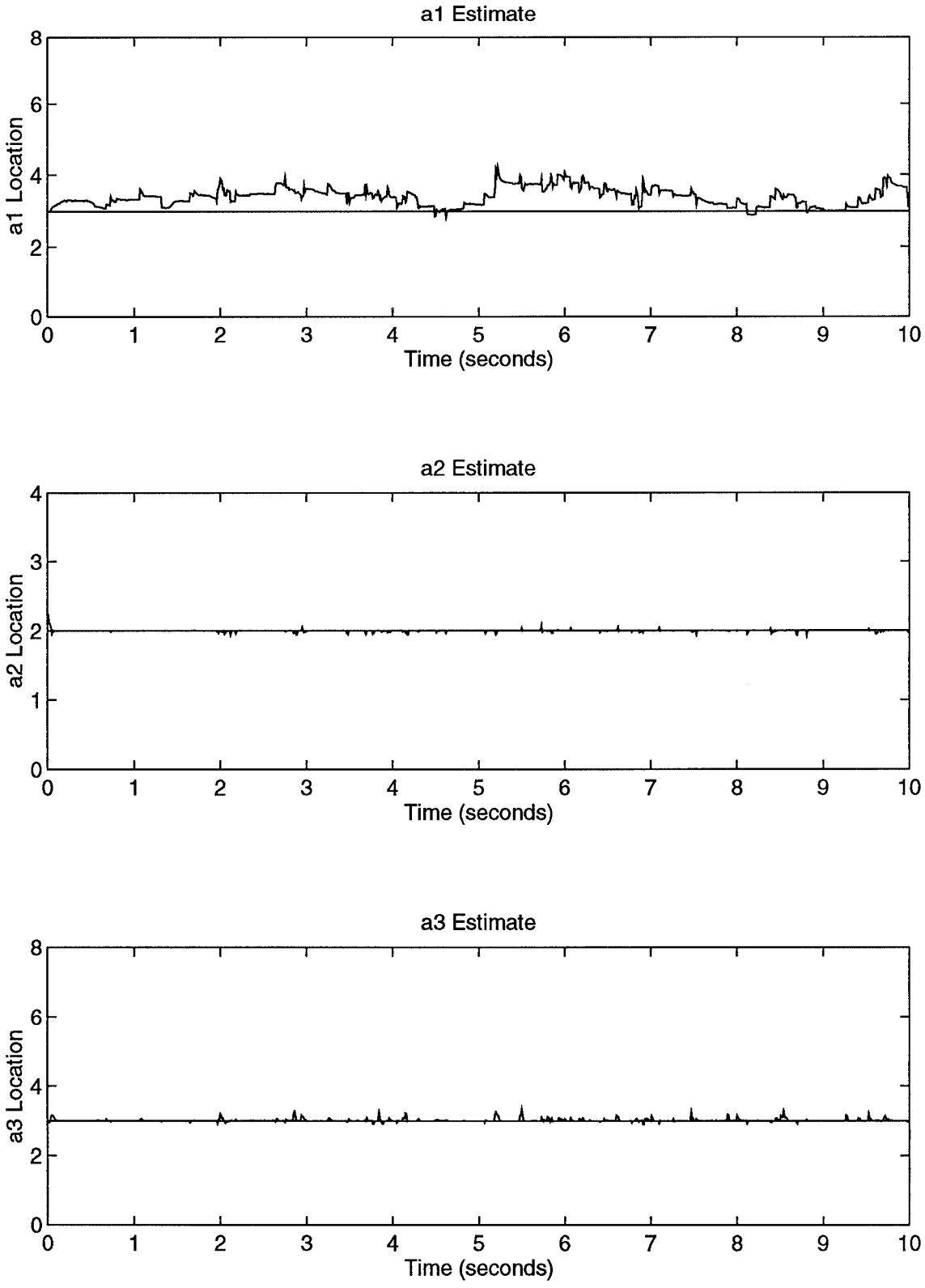


Figure A-3. Initially Matched Filter Using Residual Monitoring

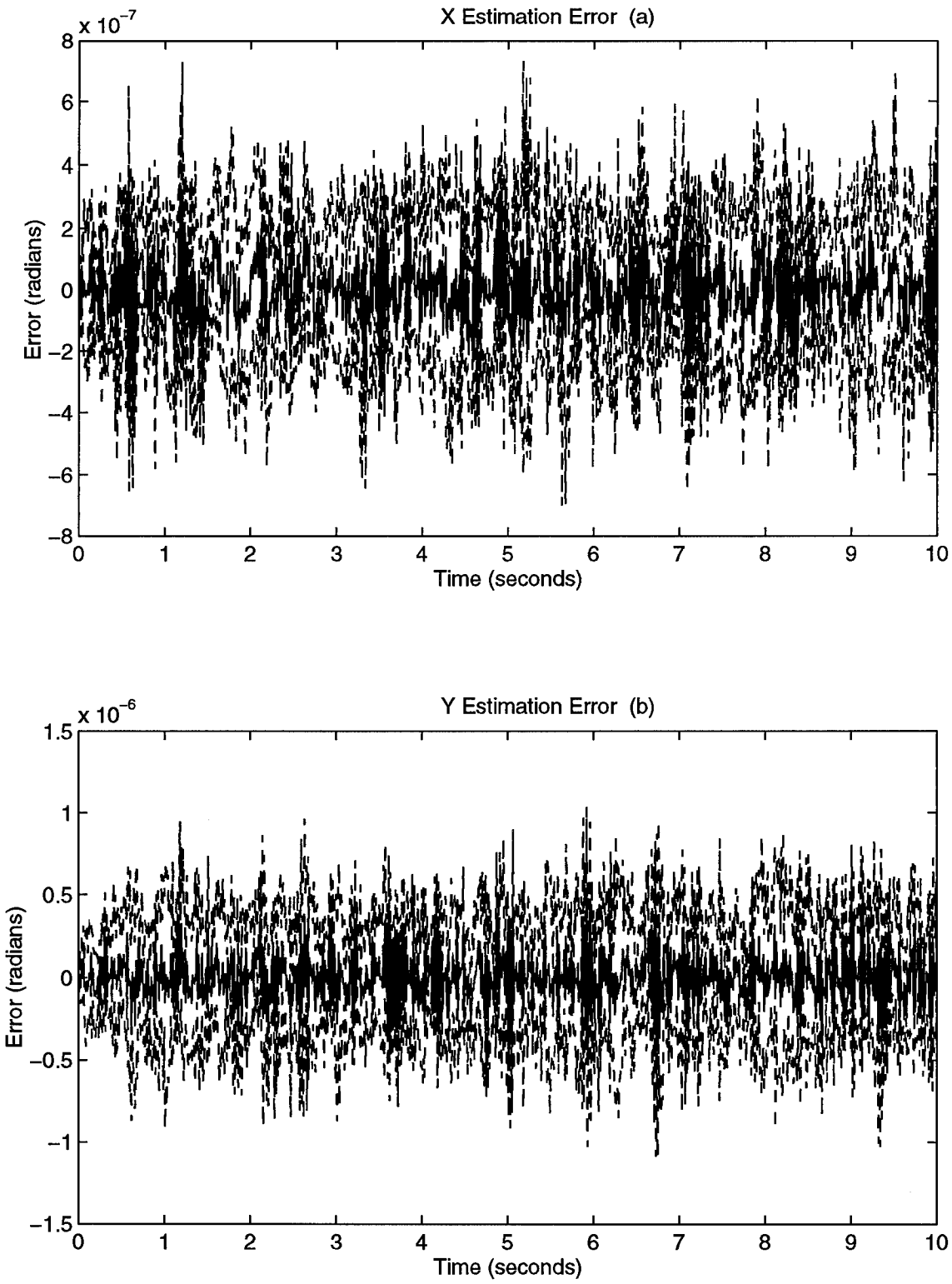
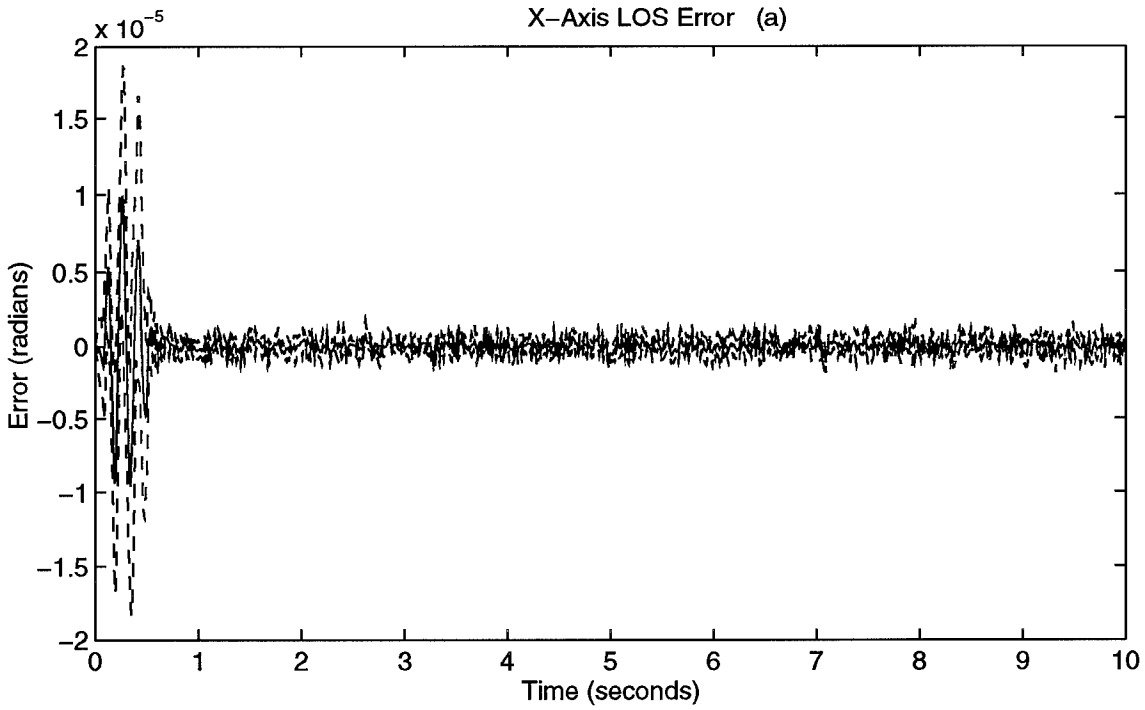


Figure A-4. Initially Matched Filter Estimation Errors



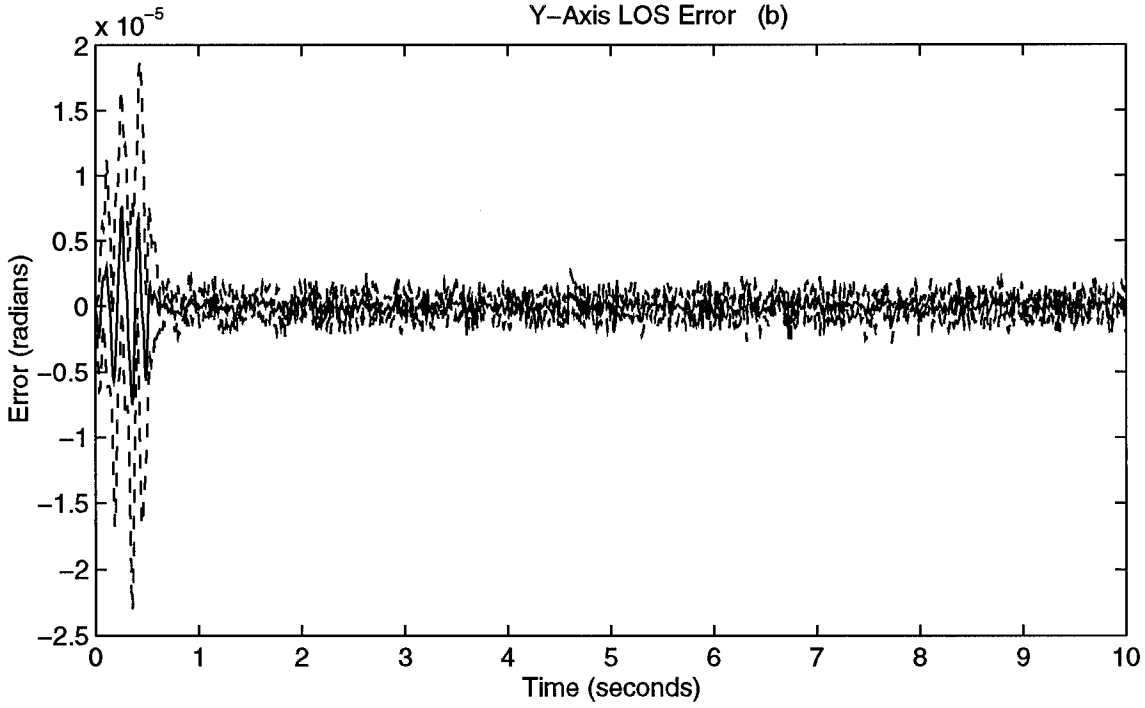


Figure A-5. Initially Matched Filter LOS Errors

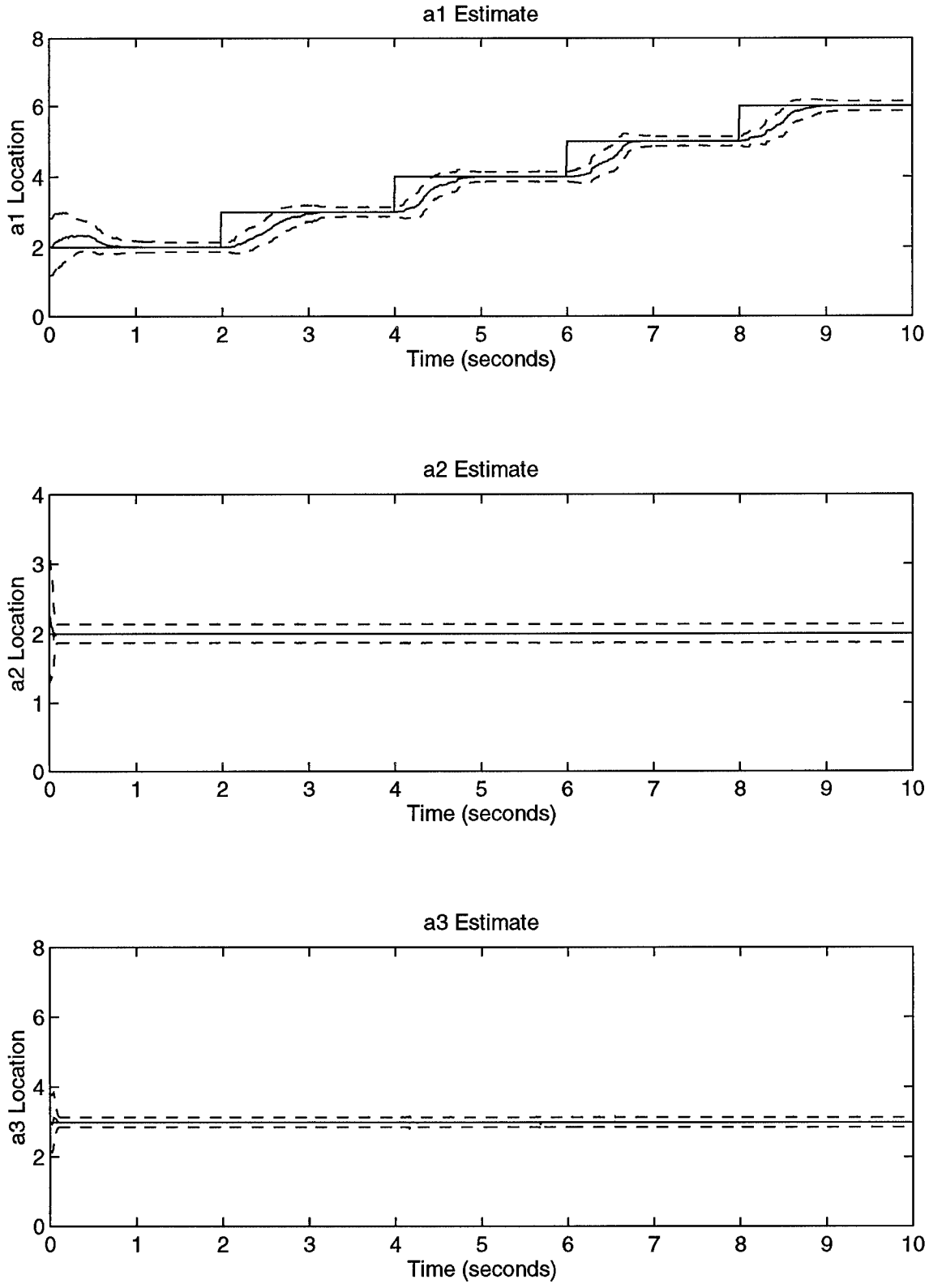


Figure A-6. a_1 Moving Up Using Parameter Position Monitoring

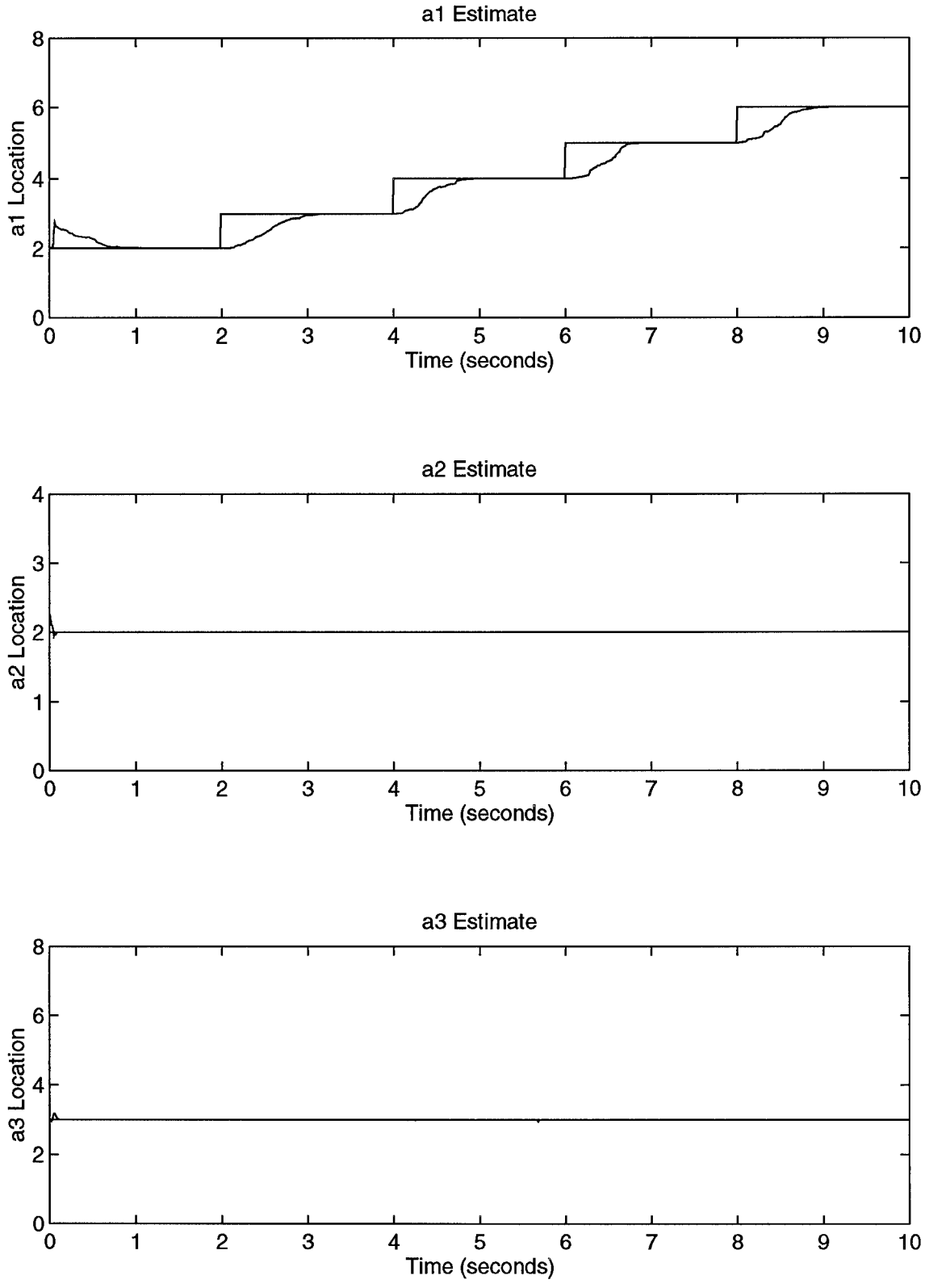


Figure A-7. a_1 Moving Up Using Probability Monitoring

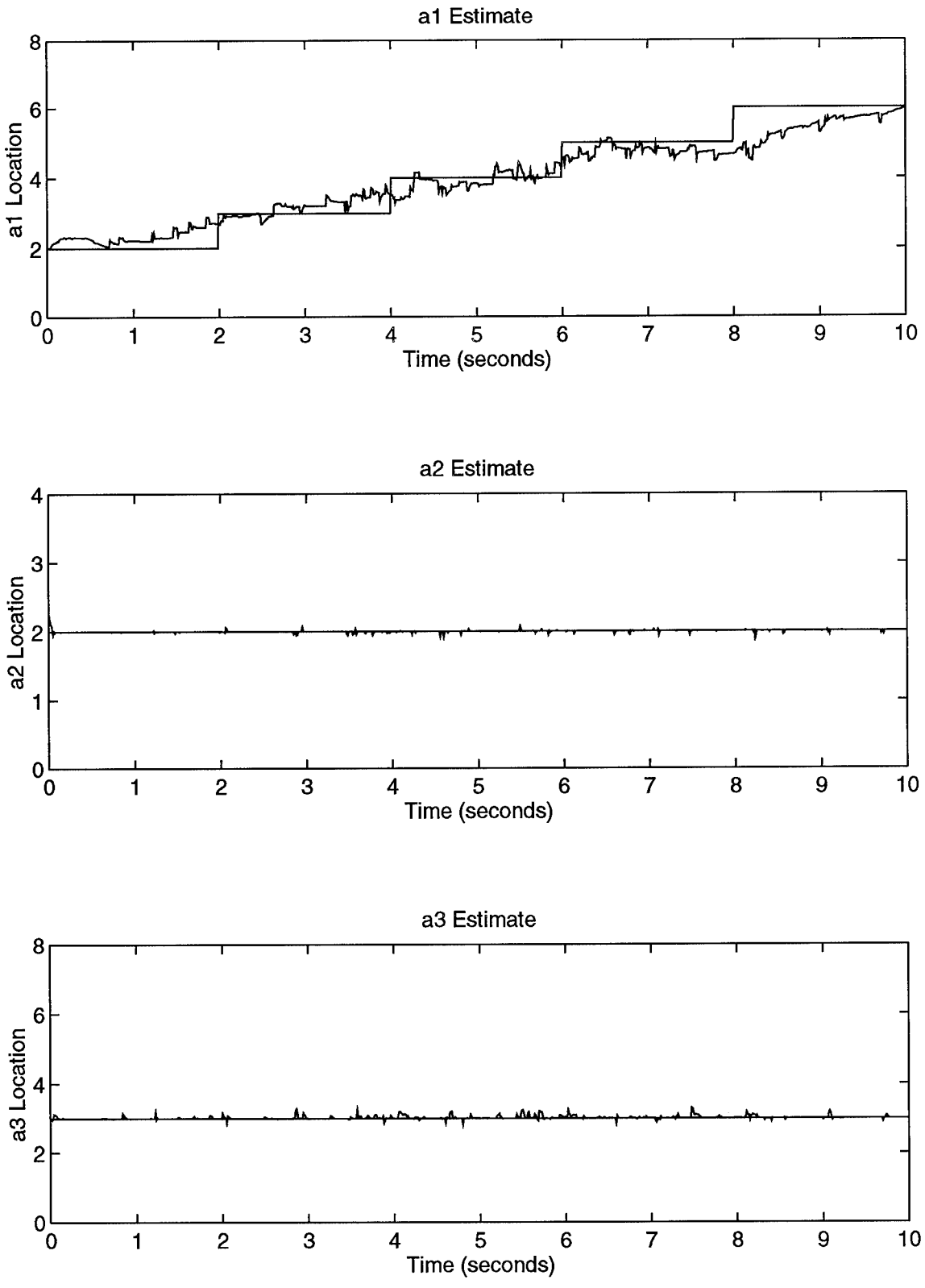
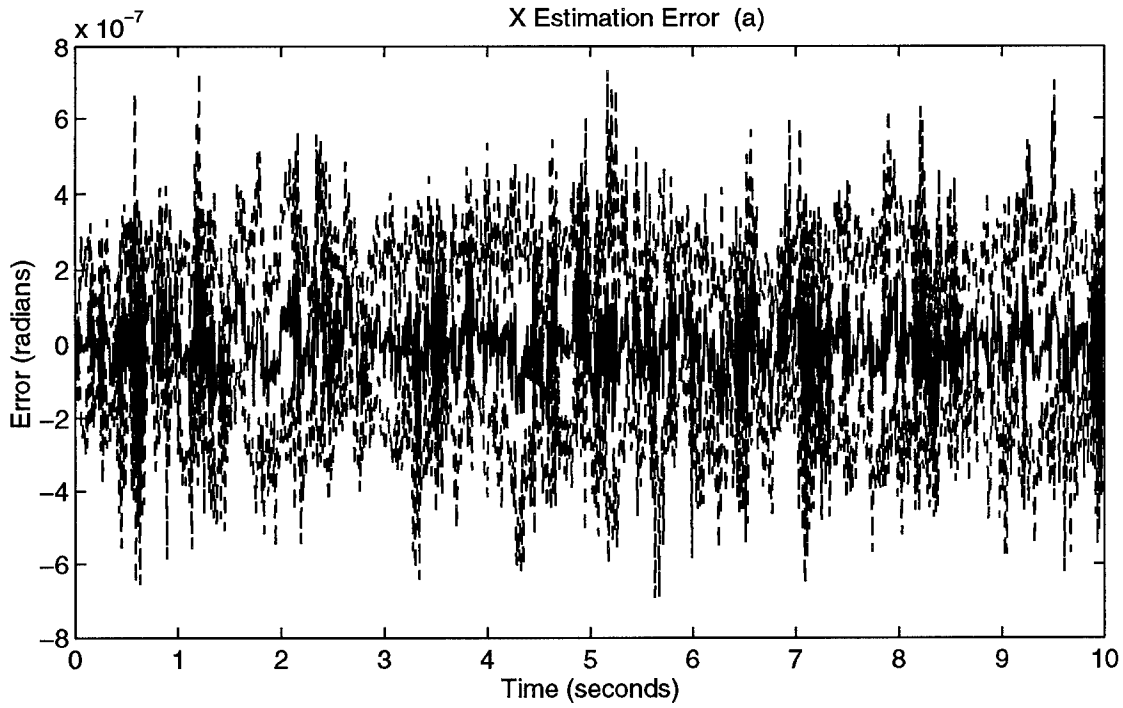


Figure A-8. a_1 Moving Up Using Residual Monitoring



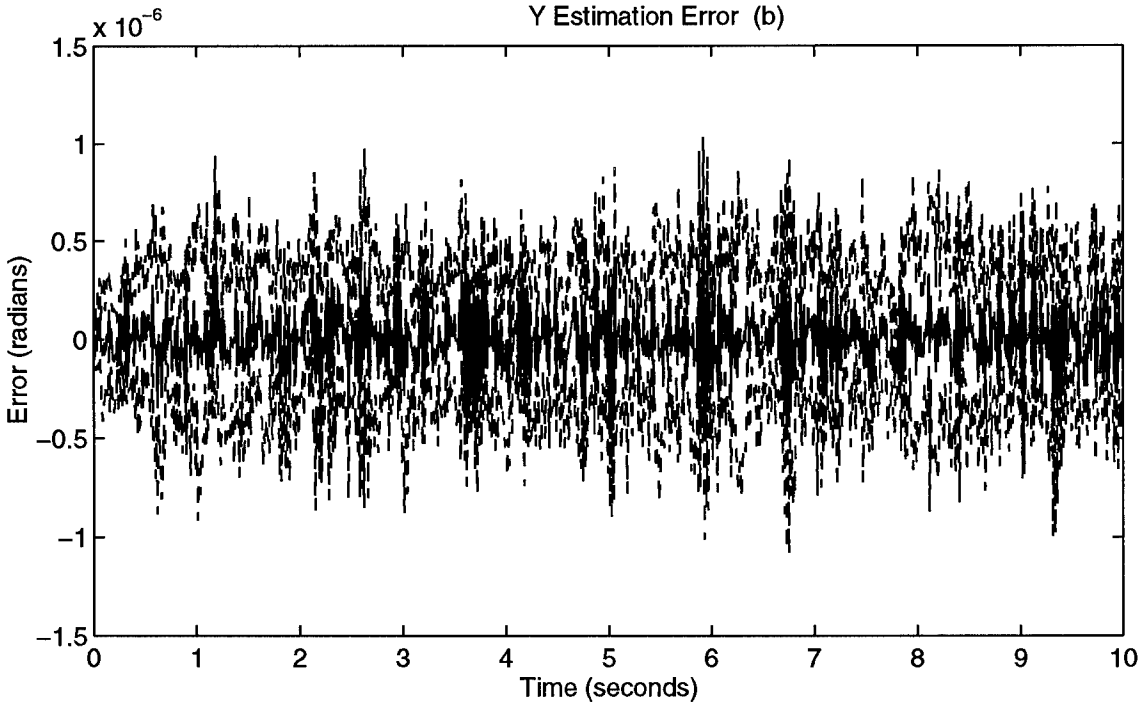
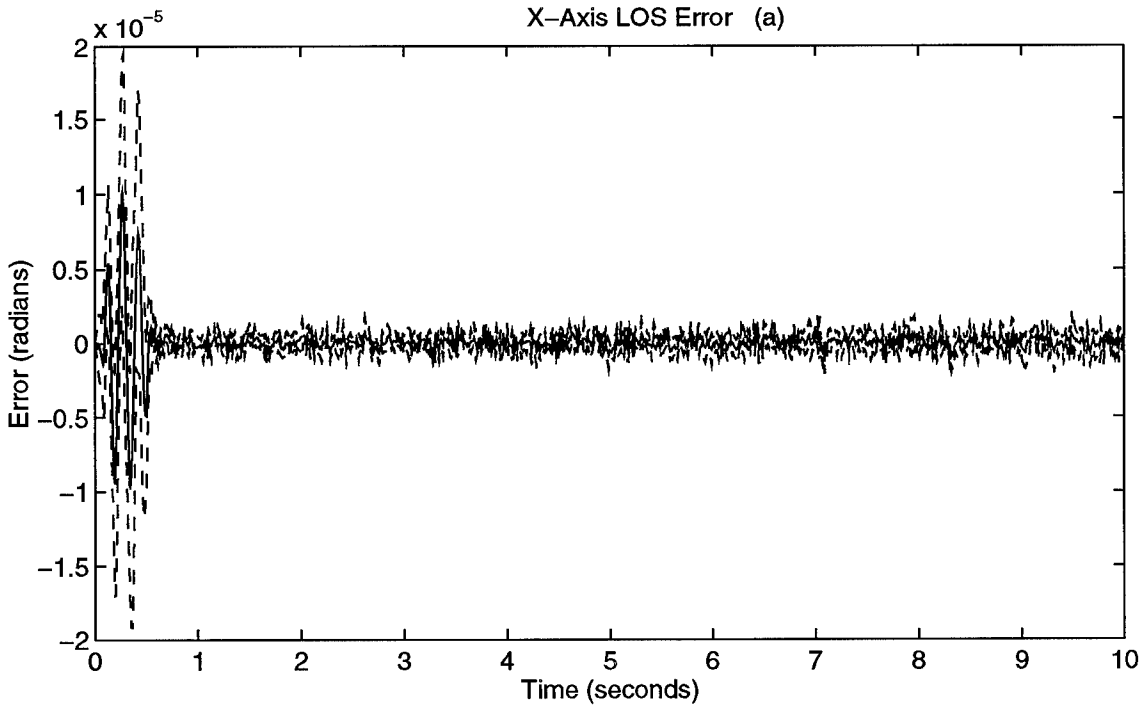


Figure A-9. a_1 Moving Up Estimation Errors



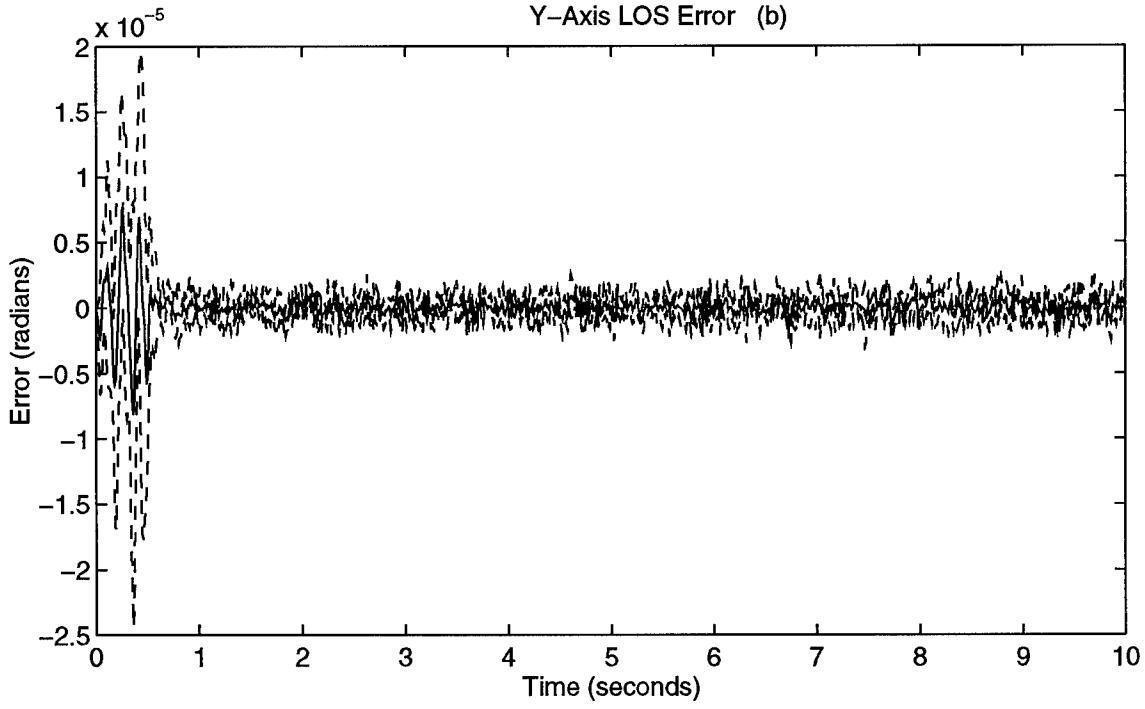


Figure A-10. a_1 Moving Up LOS Errors

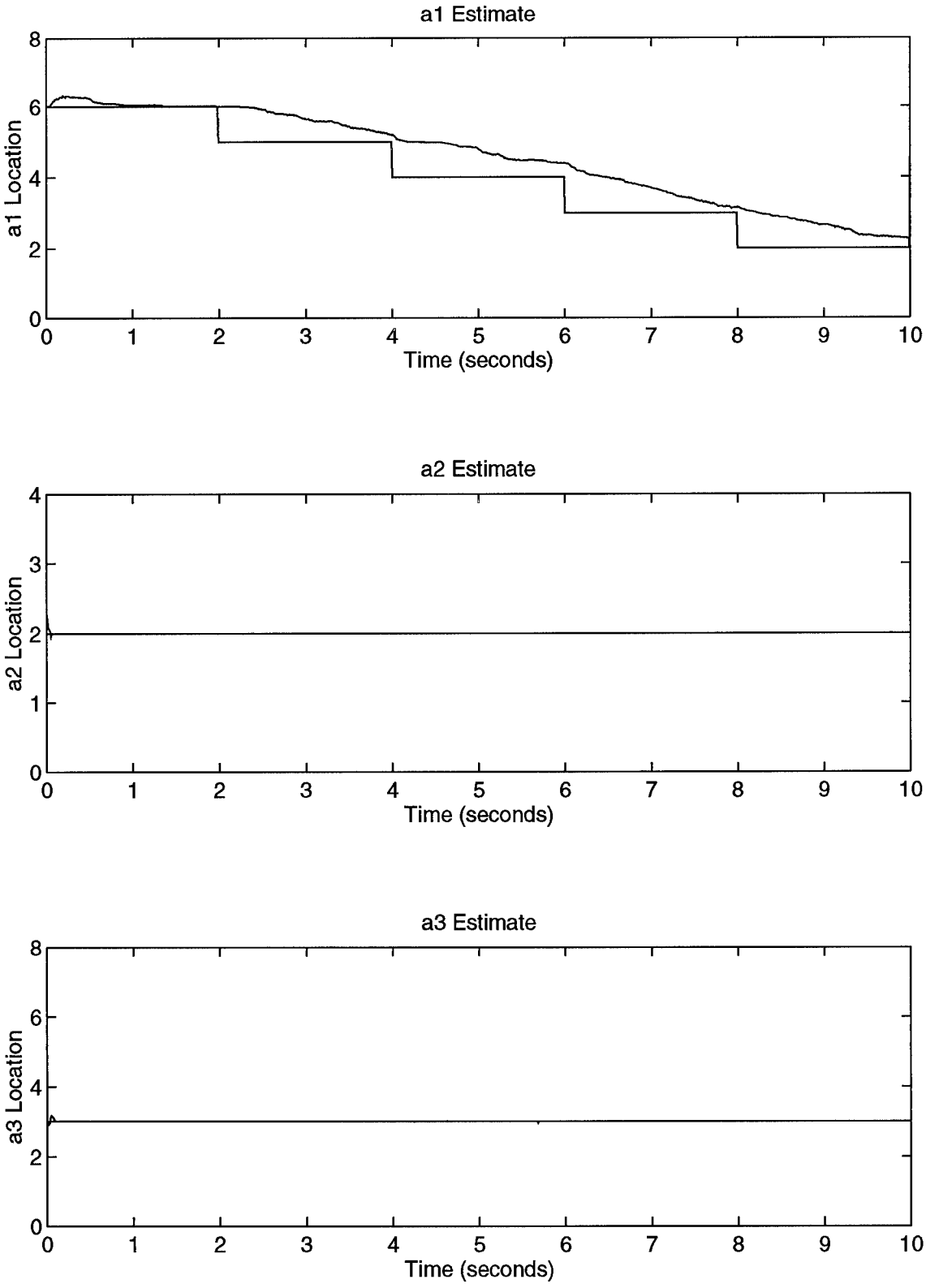


Figure A-11. a_1 Moving Down Using Parameter Position Monitoring

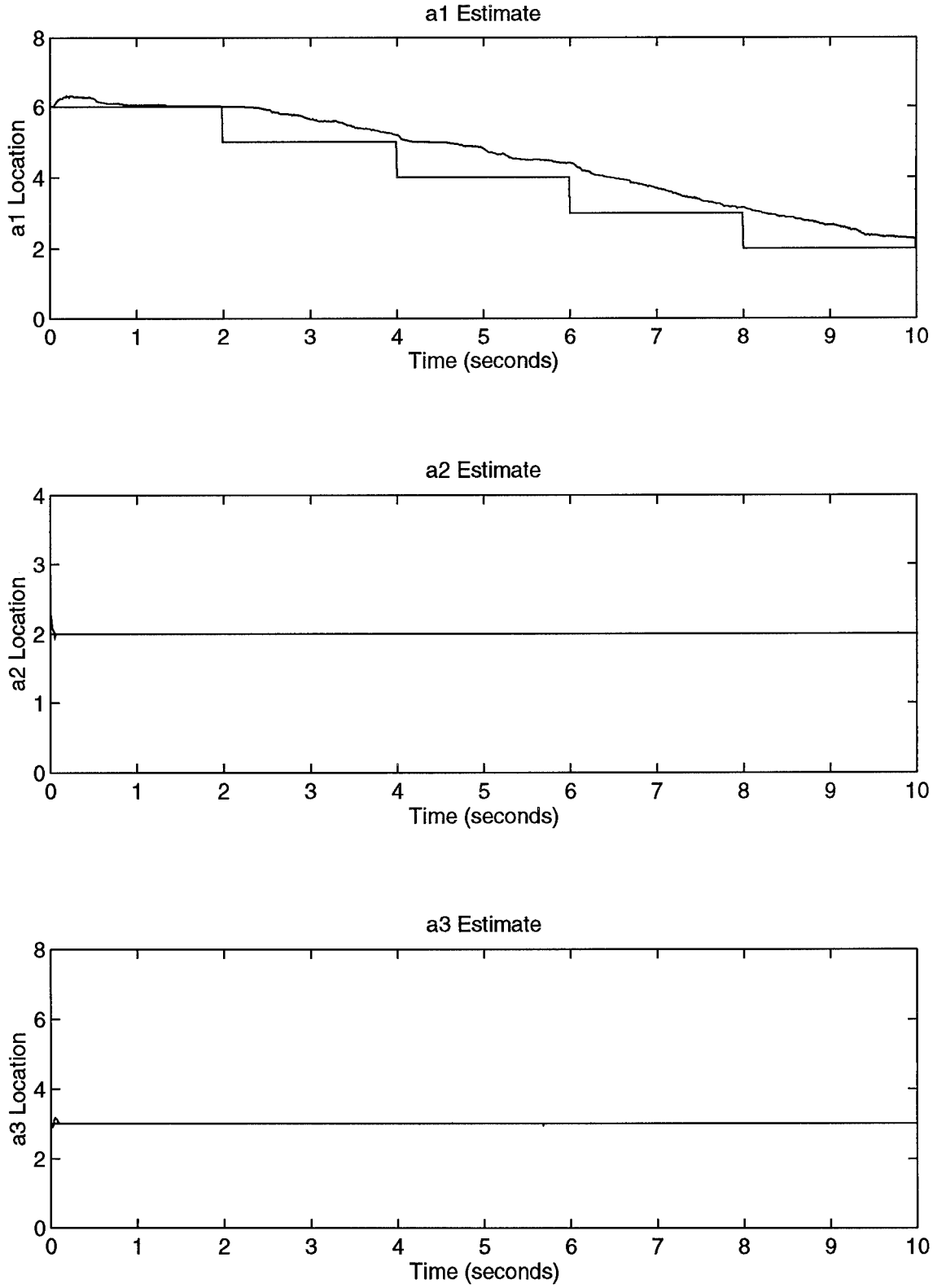


Figure A-12. a_1 Moving Down Using Probability Monitoring

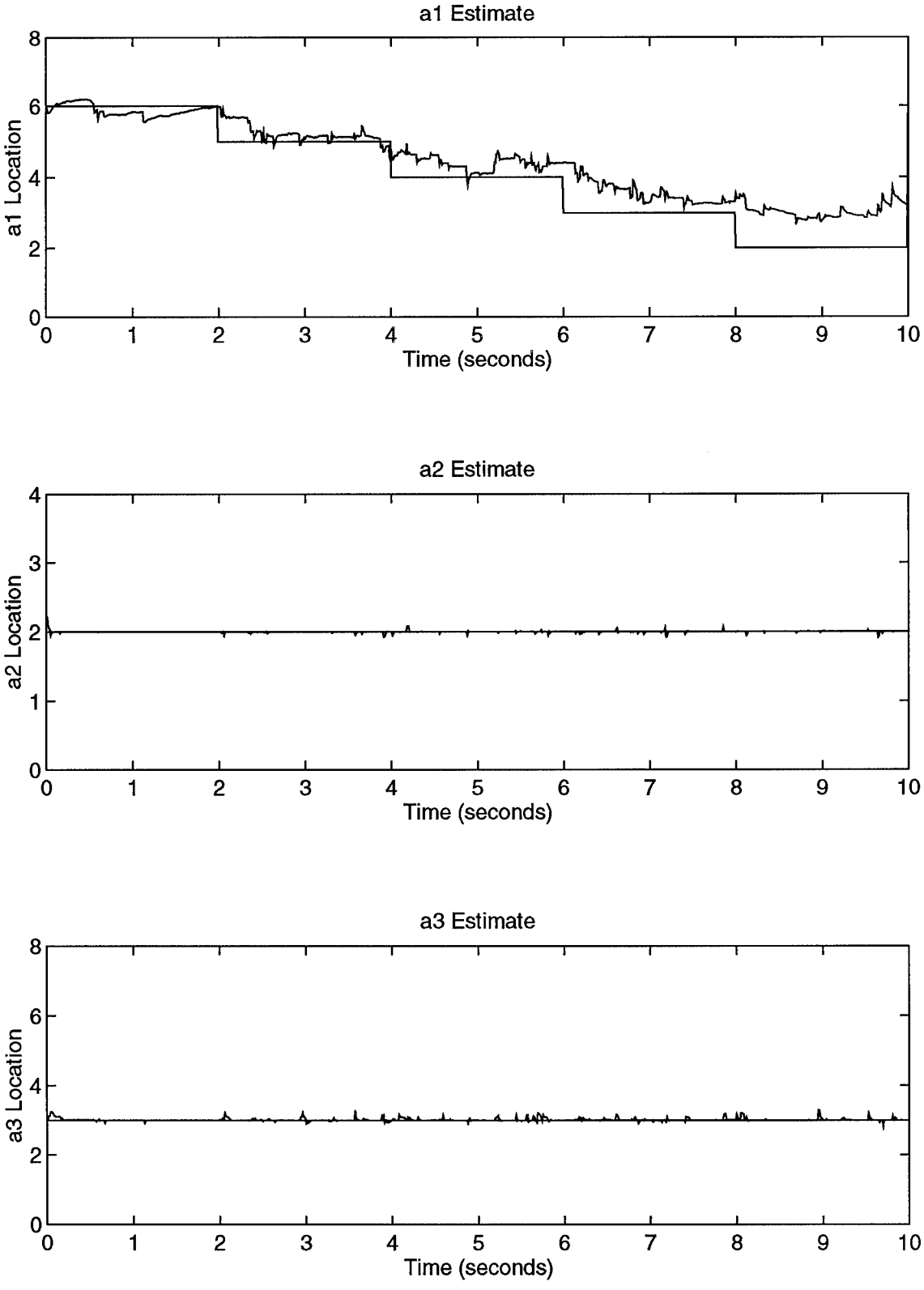
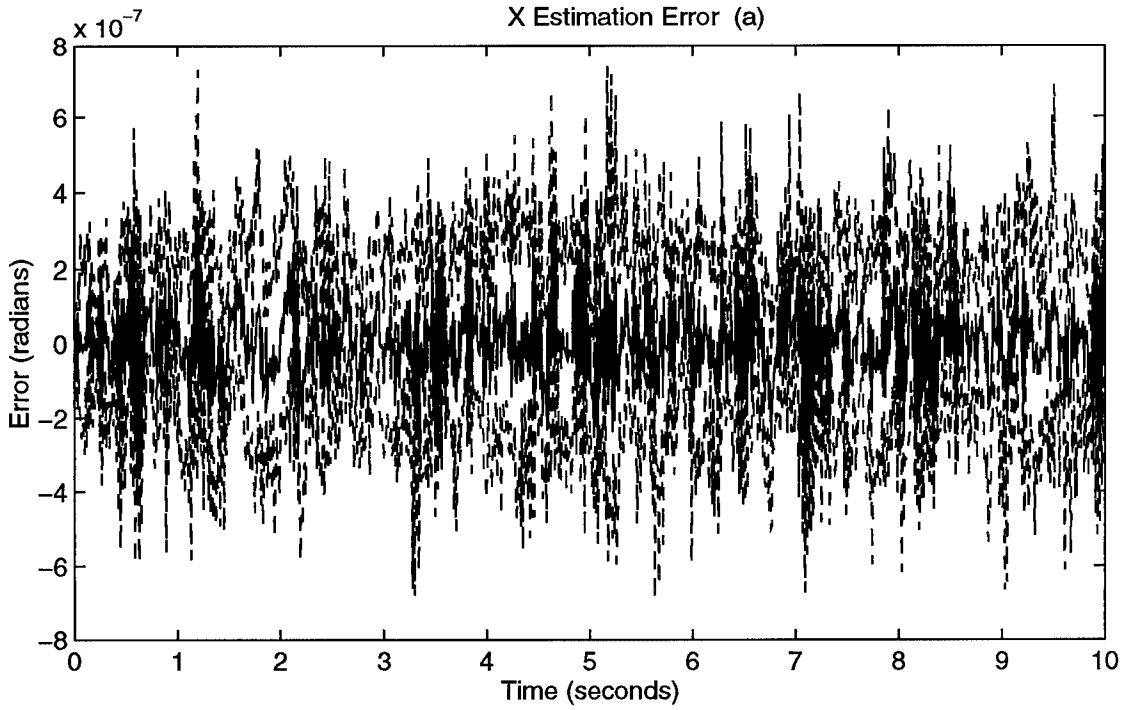


Figure A-13. a_1 Moving Down Using Residual Monitoring



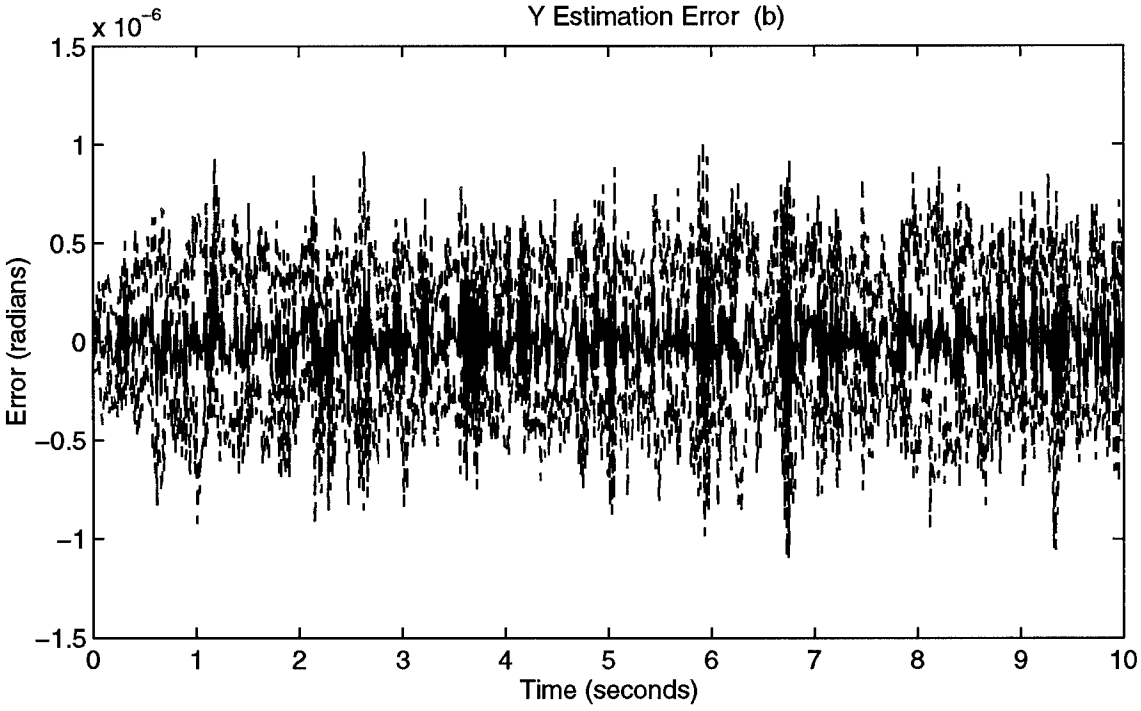
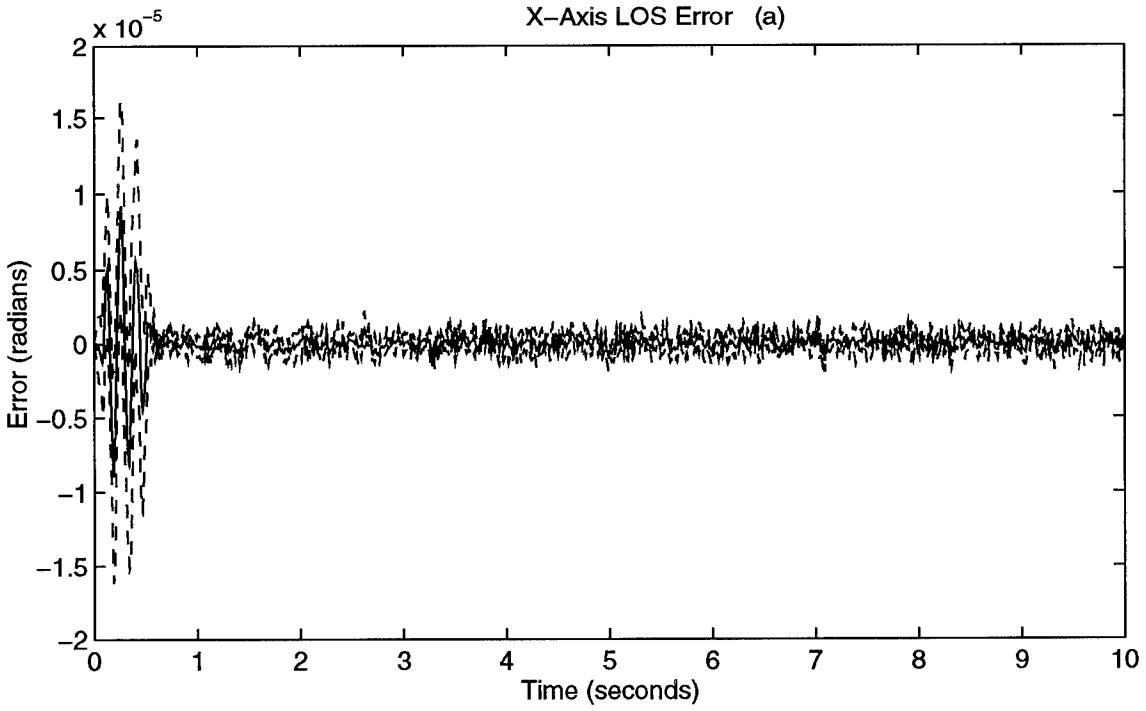


Figure A-14. a_1 Moving Down Estimation Errors



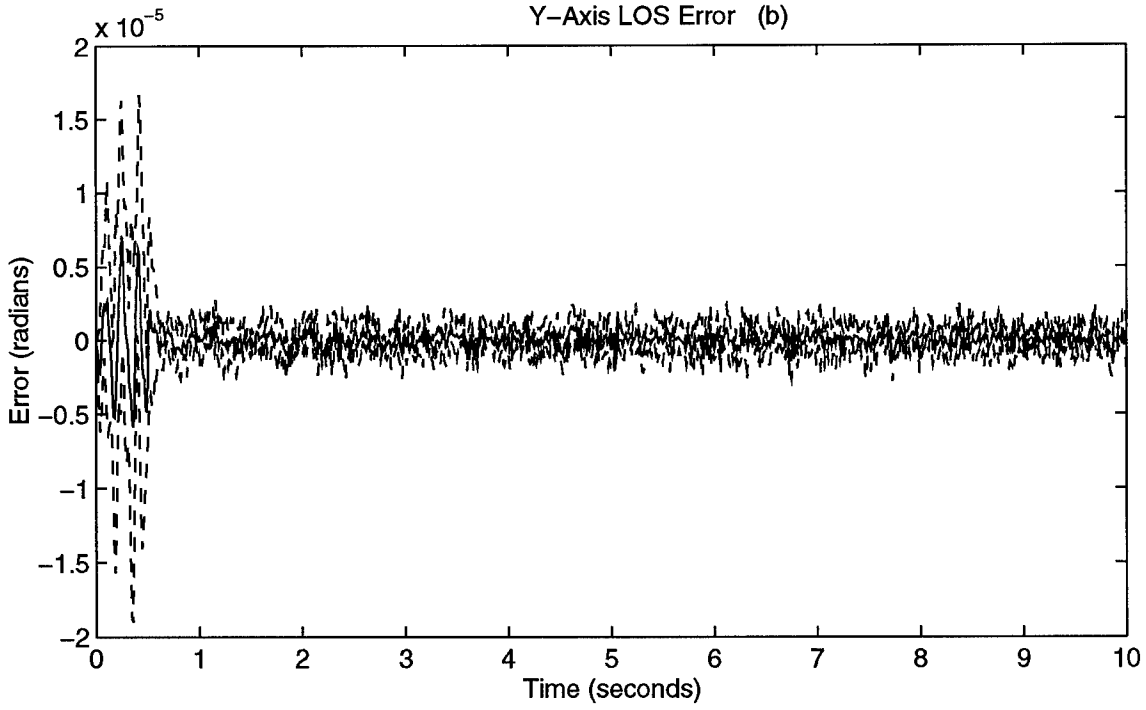


Figure A-15. a_1 Moving Down LOS Errors

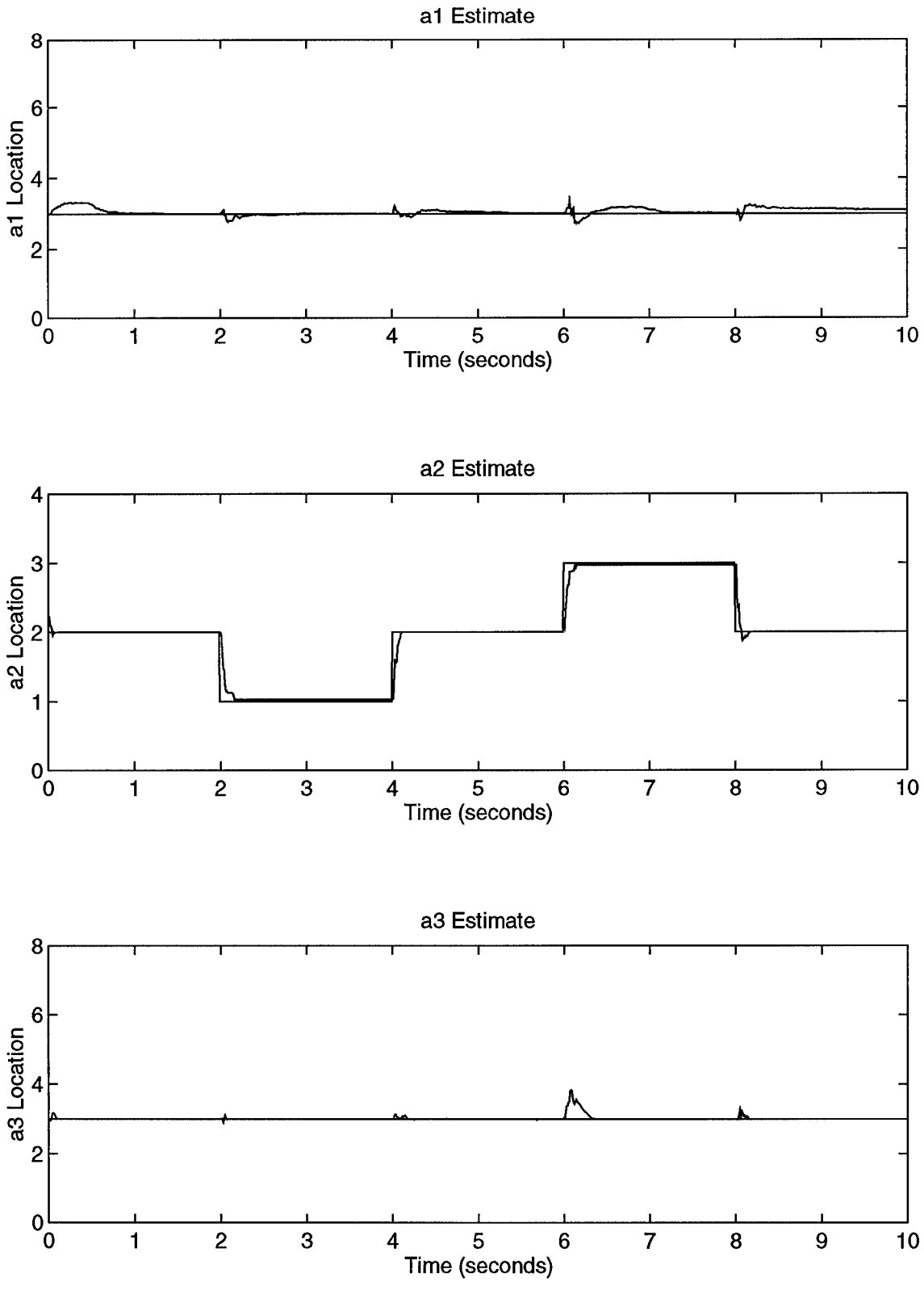


Figure A-16. a_2 Moving Using Parameter Position Monitoring

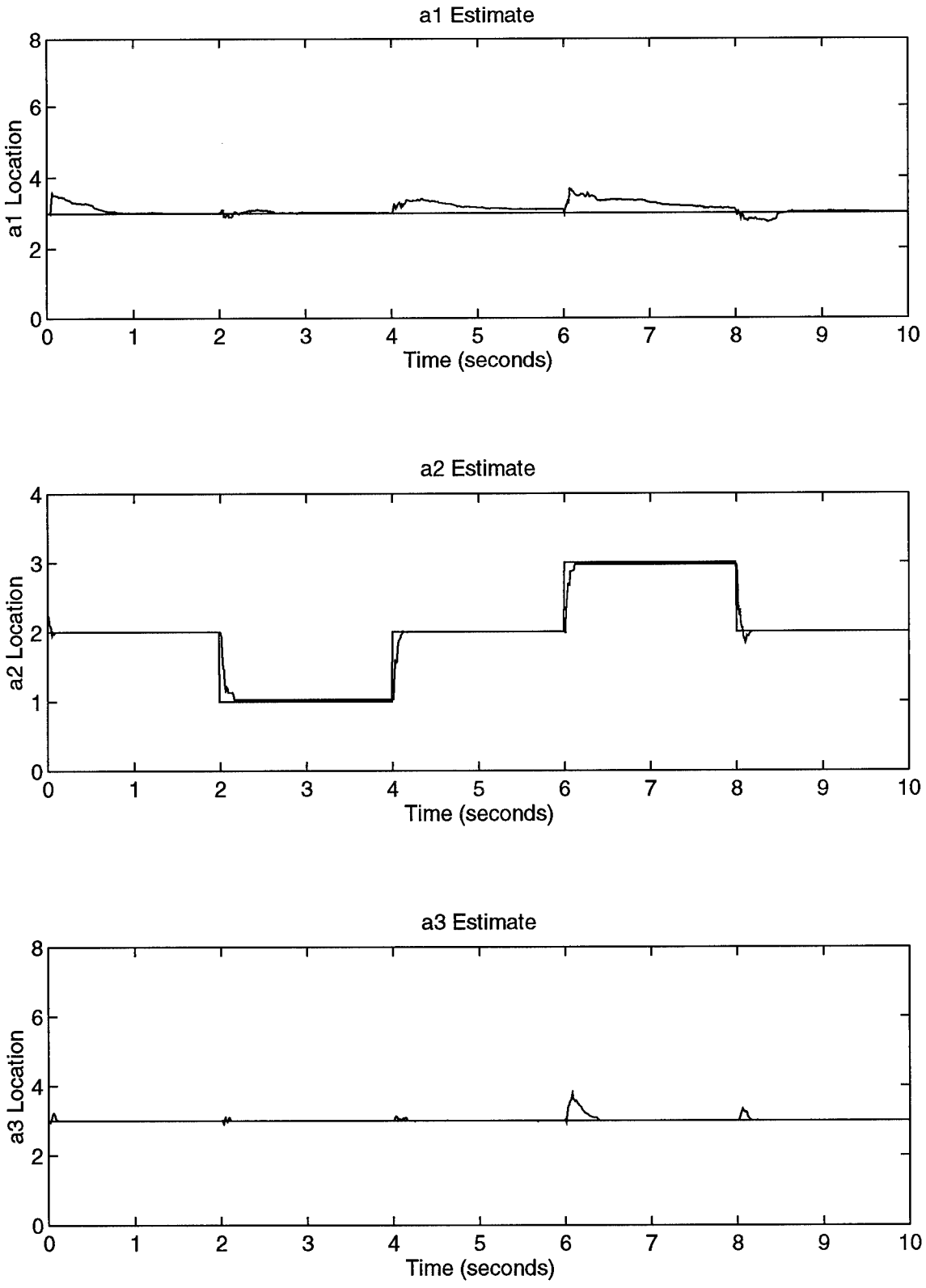


Figure A-17. a_2 Moving Using Probability Monitoring

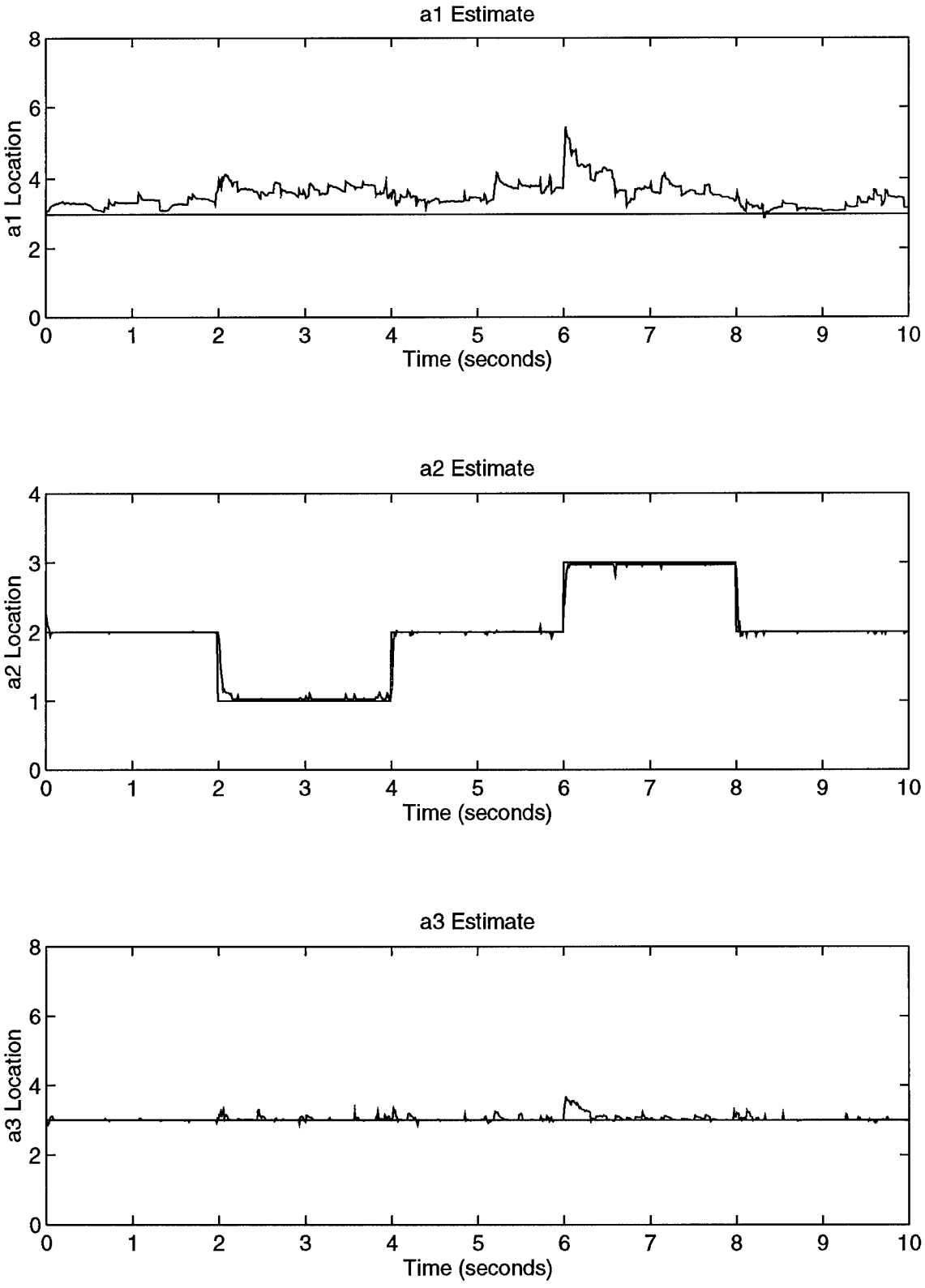
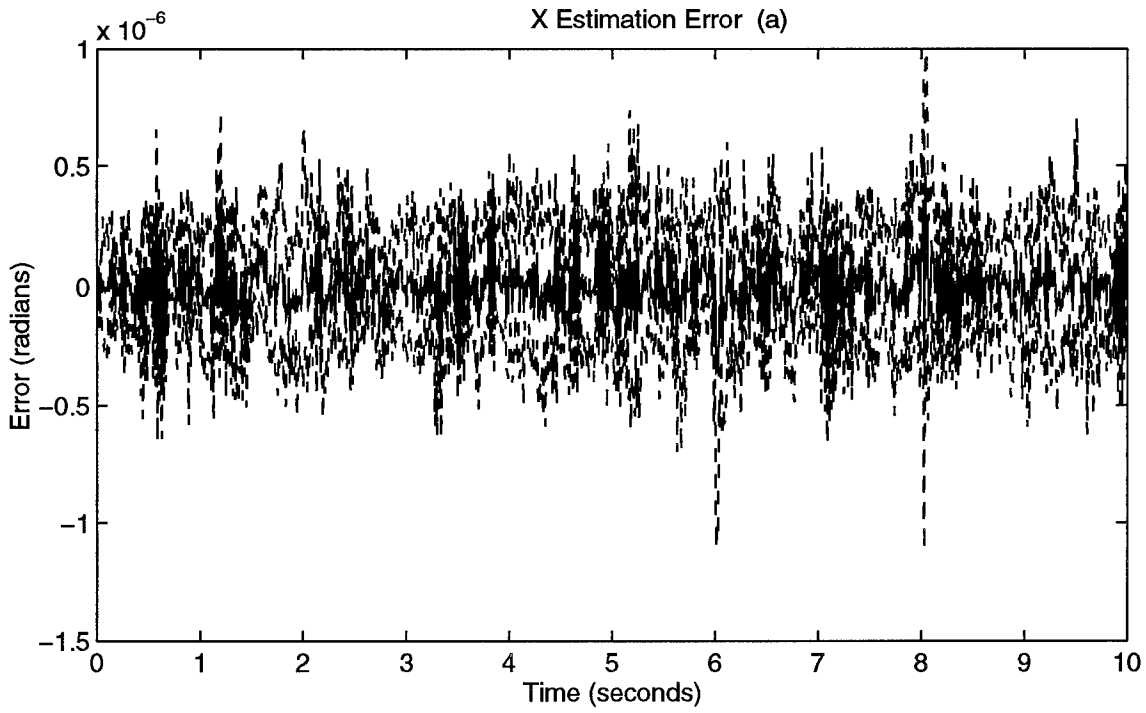


Figure A-18. a_2 Moving Using Residual Monitoring



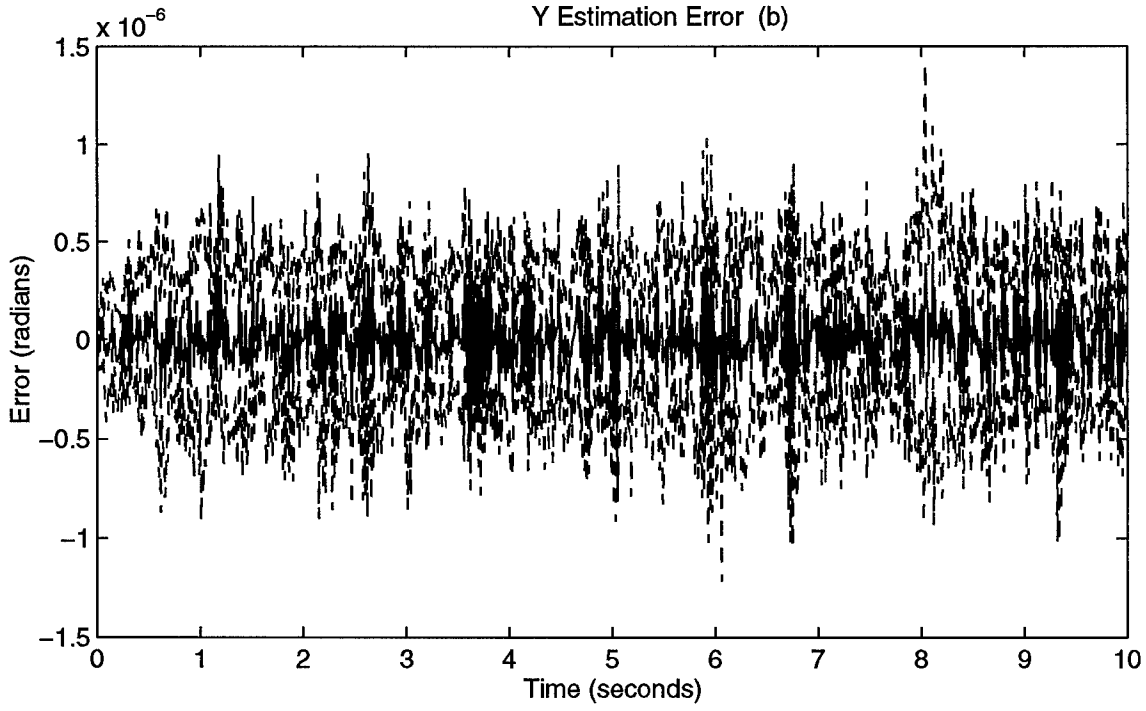
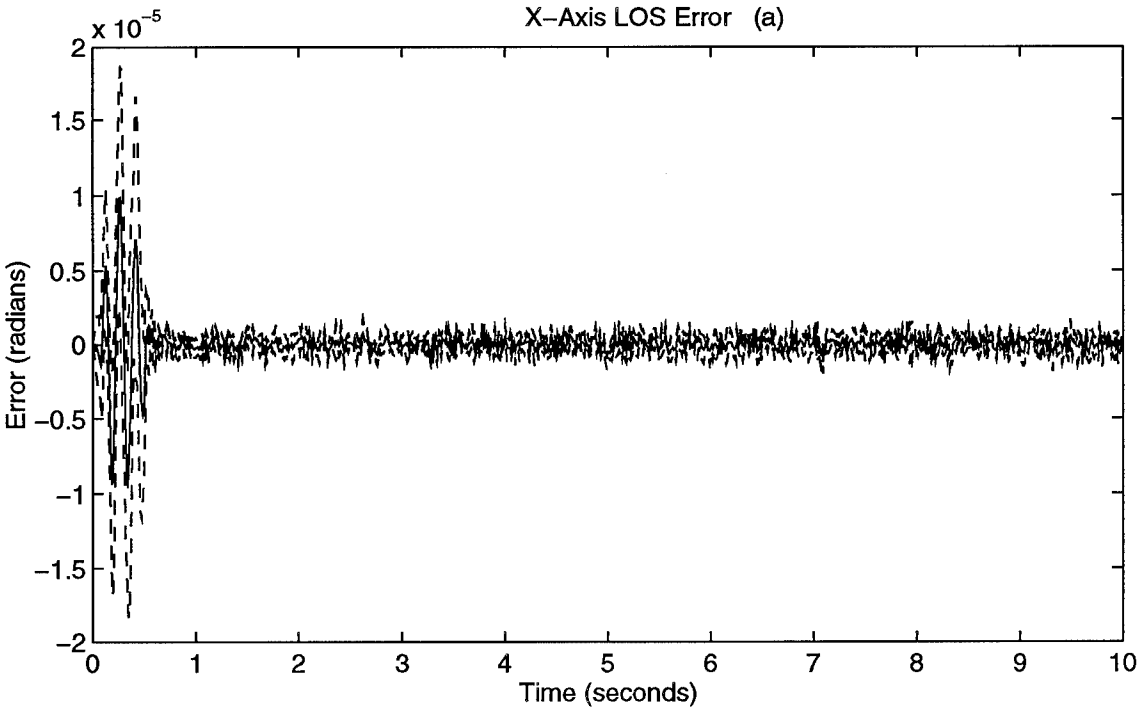


Figure A-19. a_2 Moving Estimation Errors



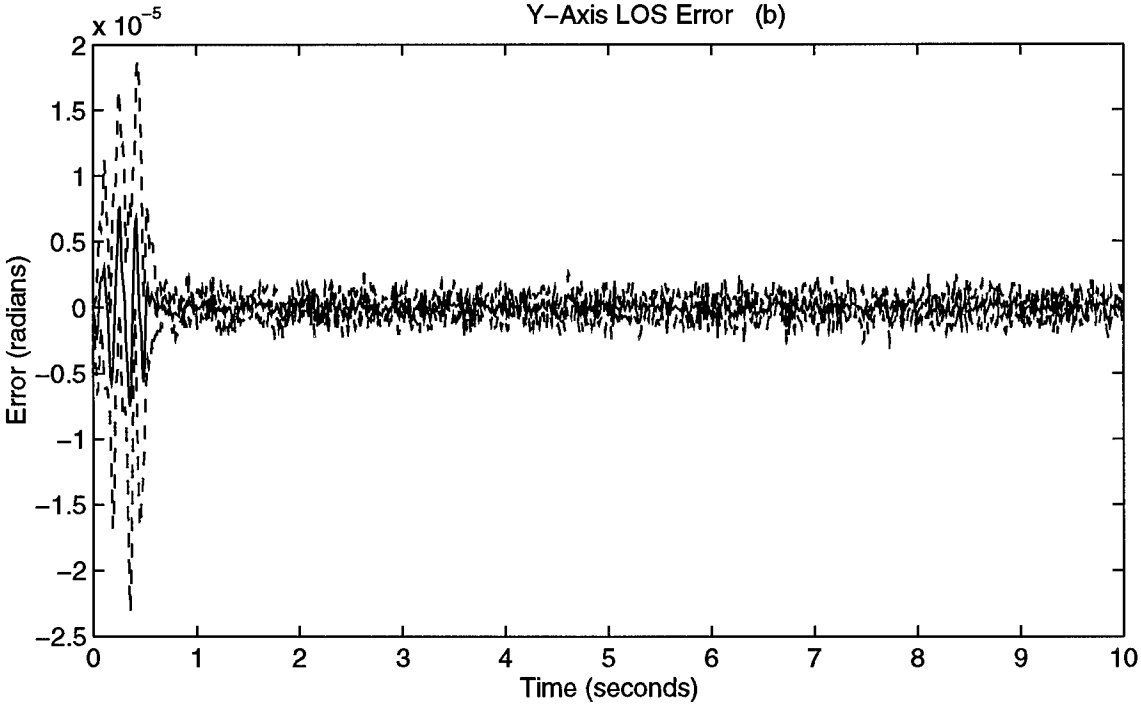


Figure A-20. a_2 Moving LOS Errors

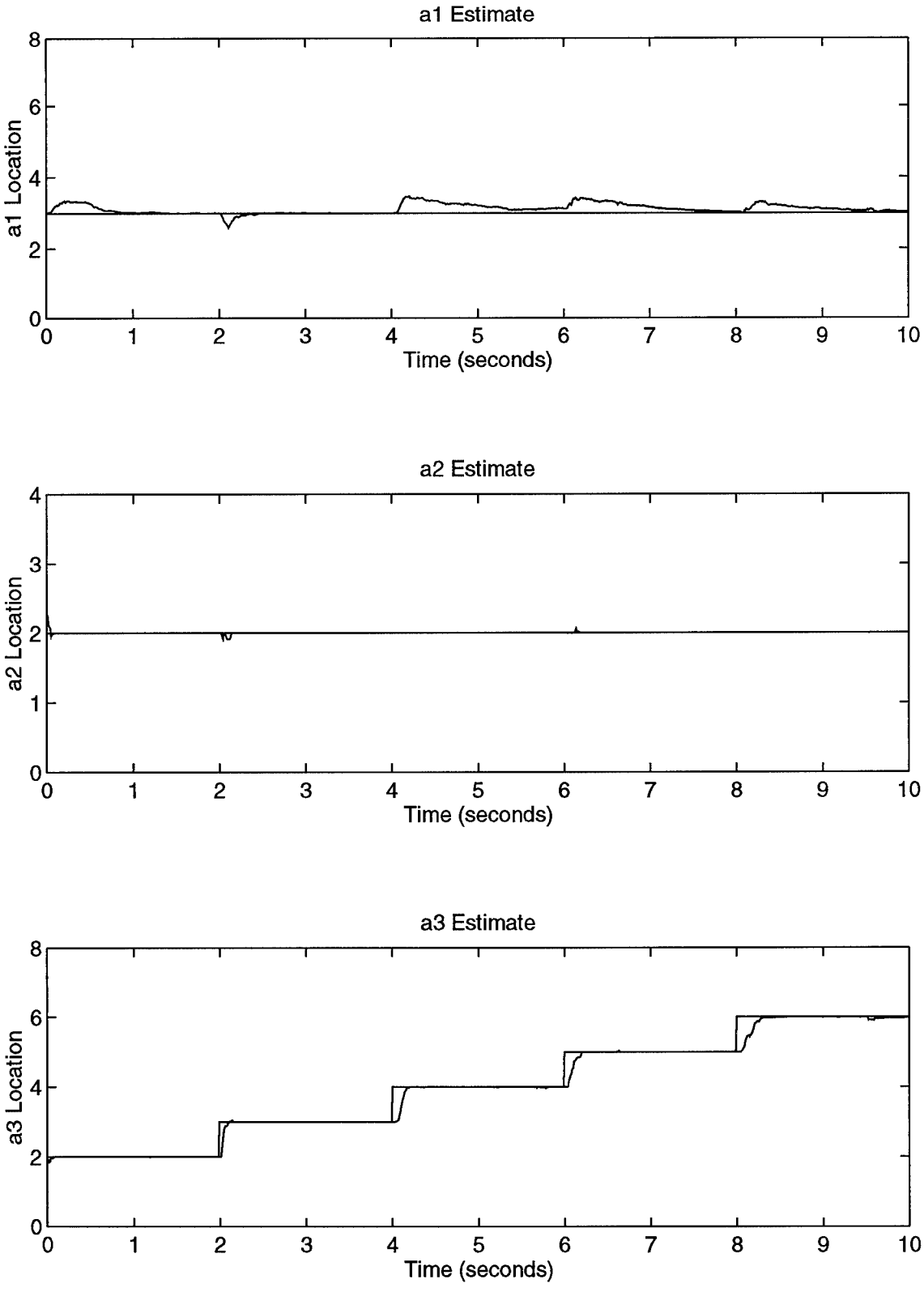


Figure A-21. a_3 Moving Up Using Parameter Position Monitoring

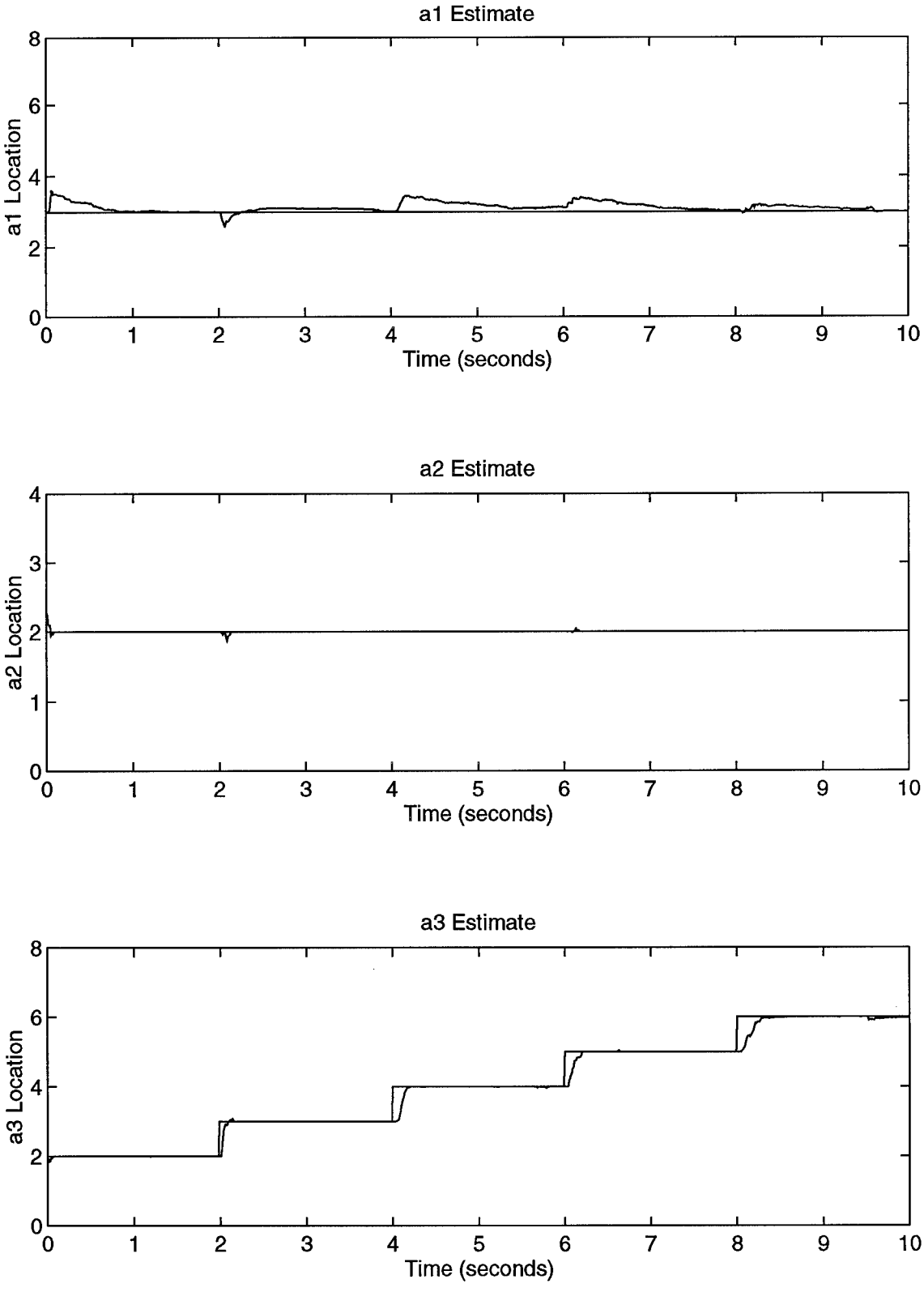


Figure A-22. a_3 Moving Up Using Probability Monitoring

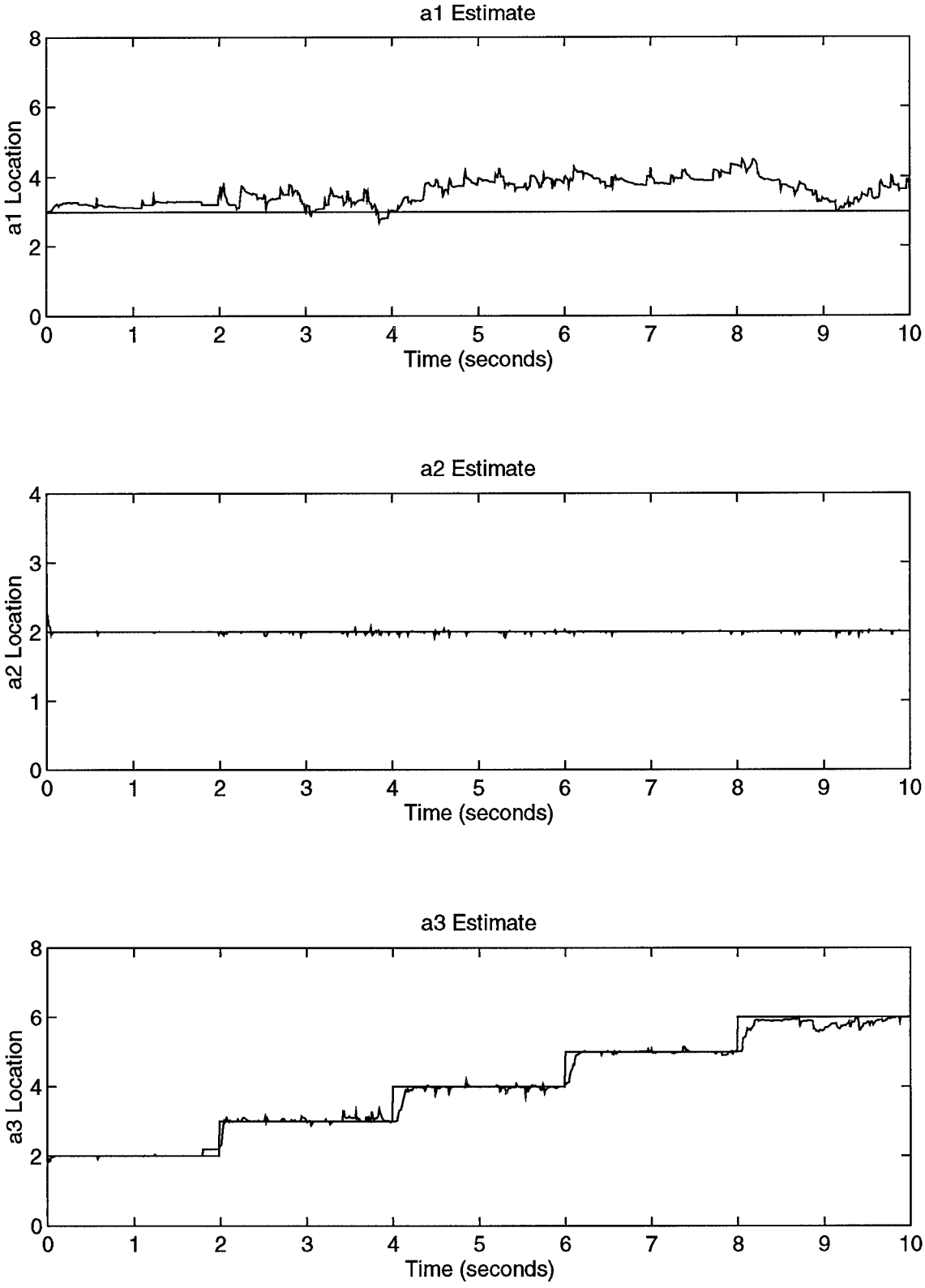
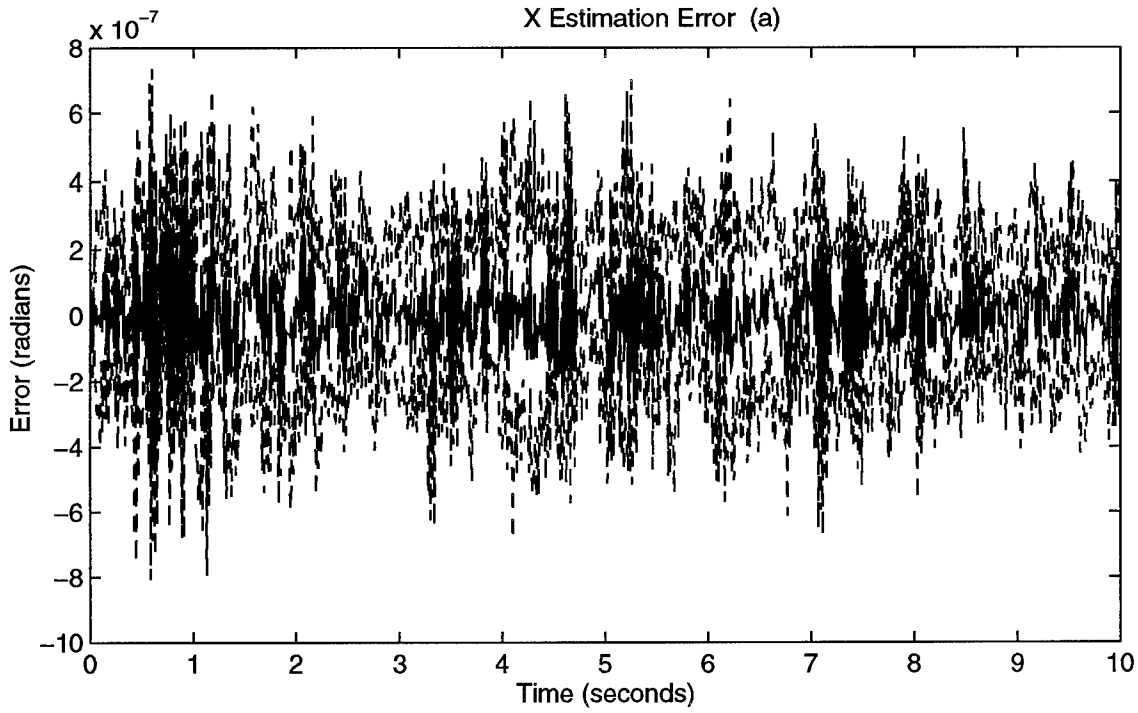


Figure A-23. a_3 Moving Up Using Residual Monitoring



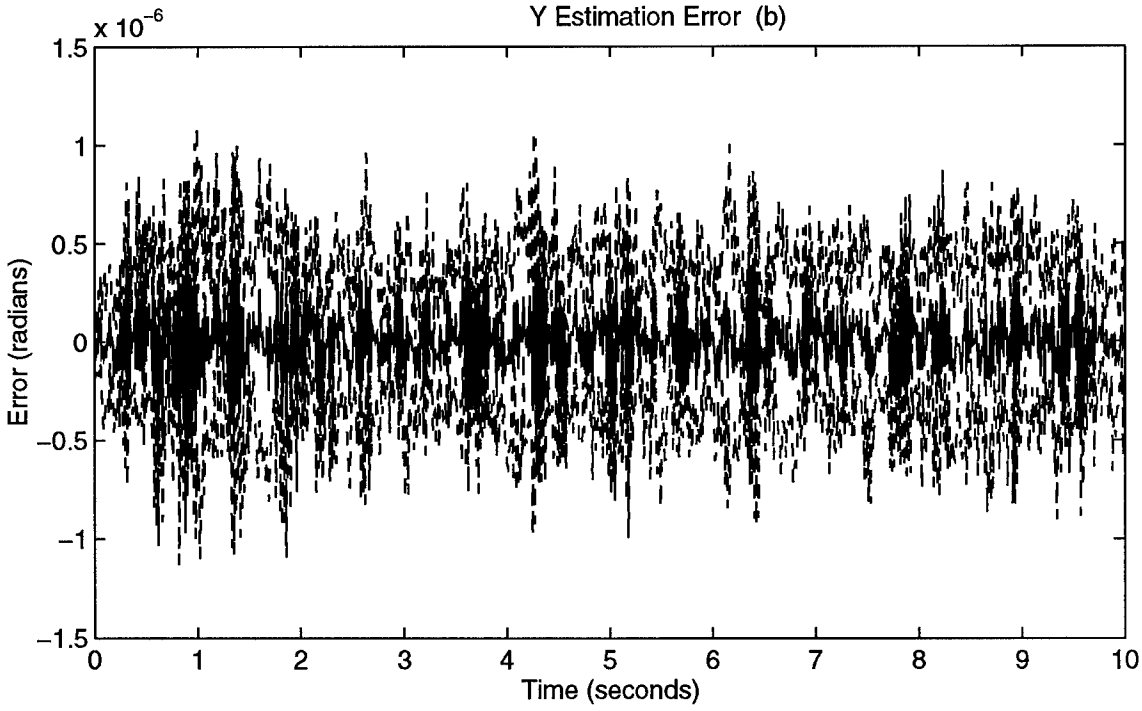
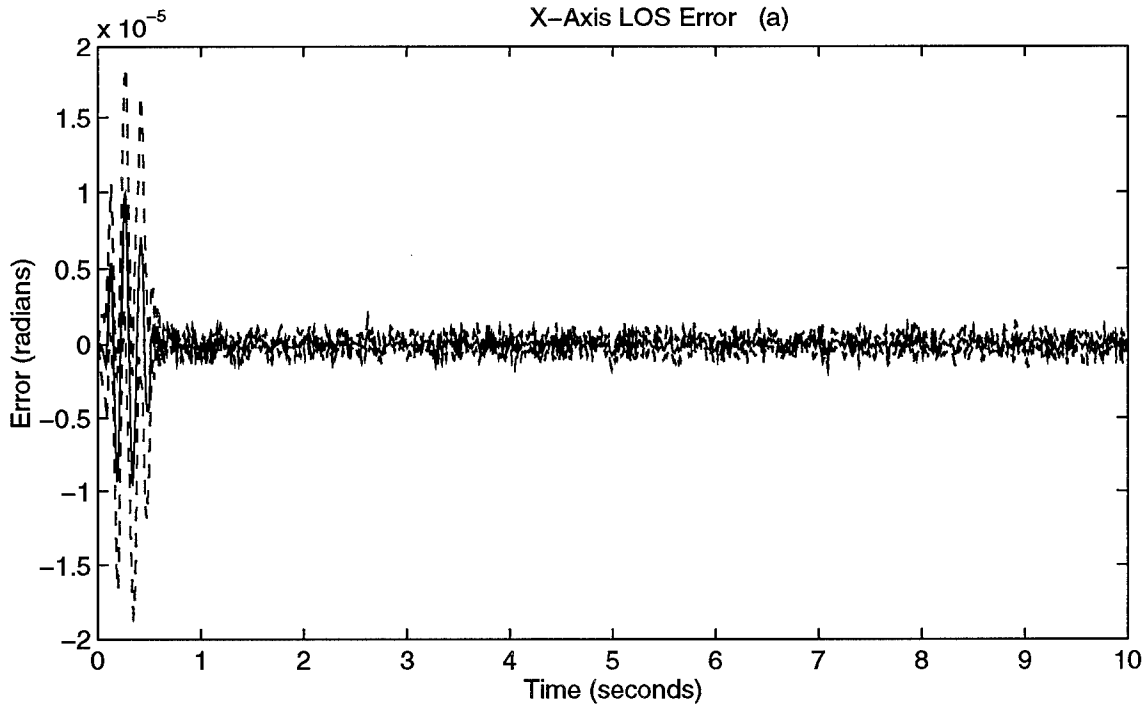


Figure A-24. a_3 Moving Up Estimation Errors



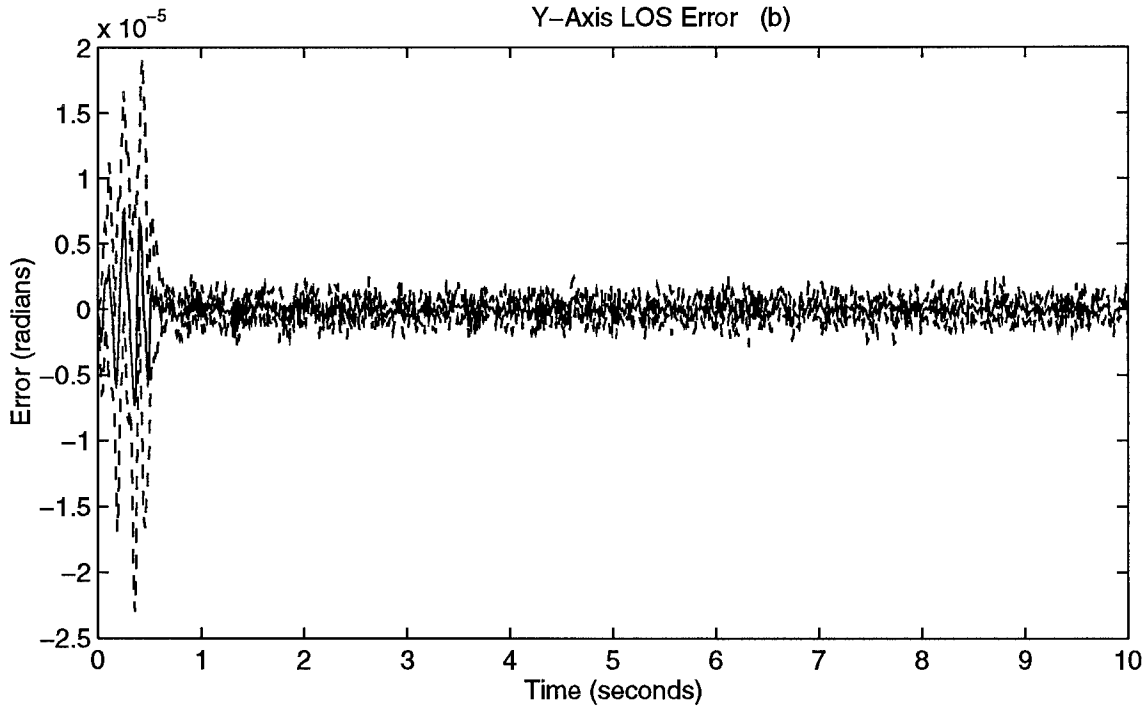


Figure A-25. a_3 Moving Up LOS Errors

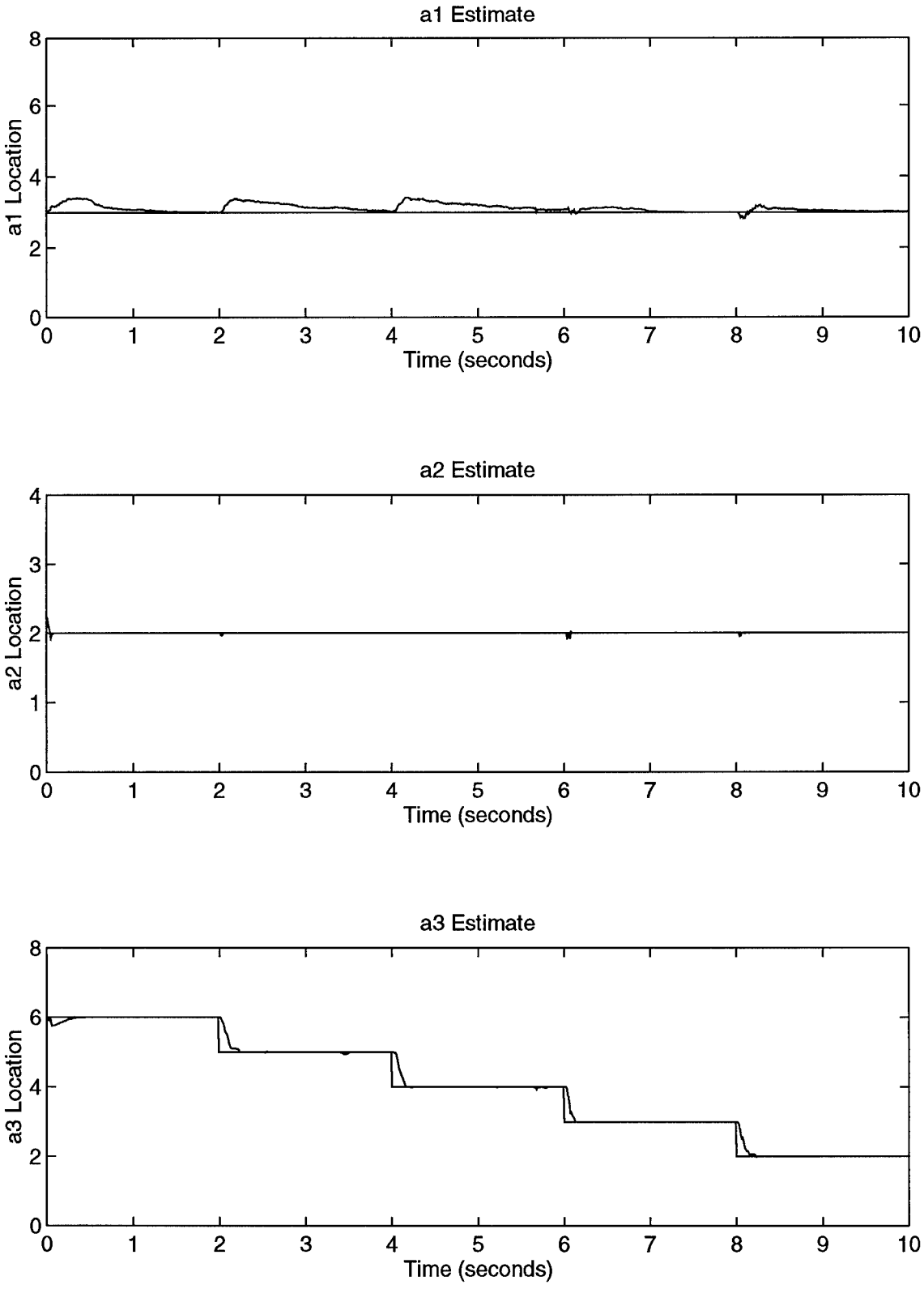


Figure A-26. a_3 Moving Down Using Parameter Position Monitoring

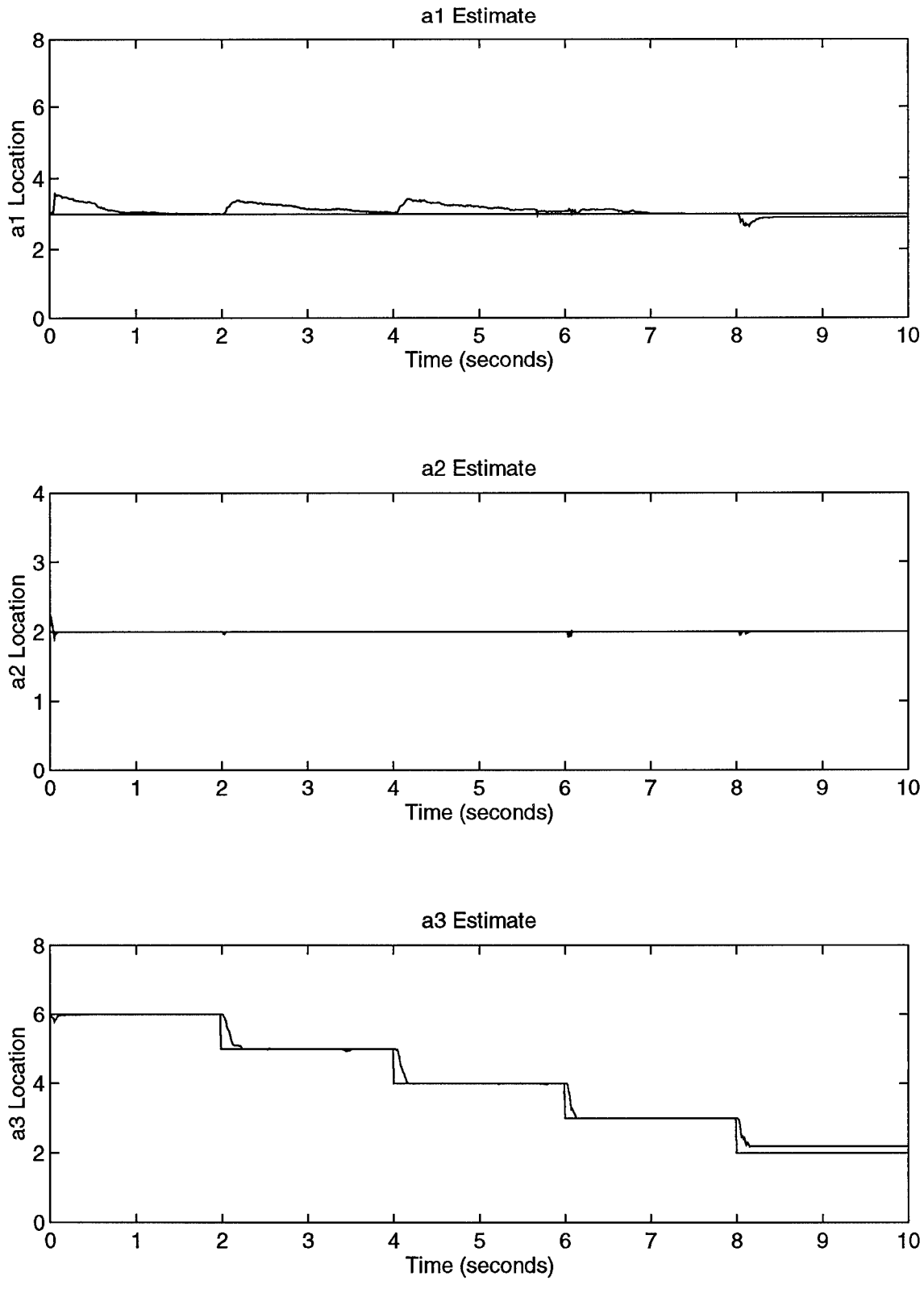


Figure A-27. a_3 Moving Down Using Probability Monitoring

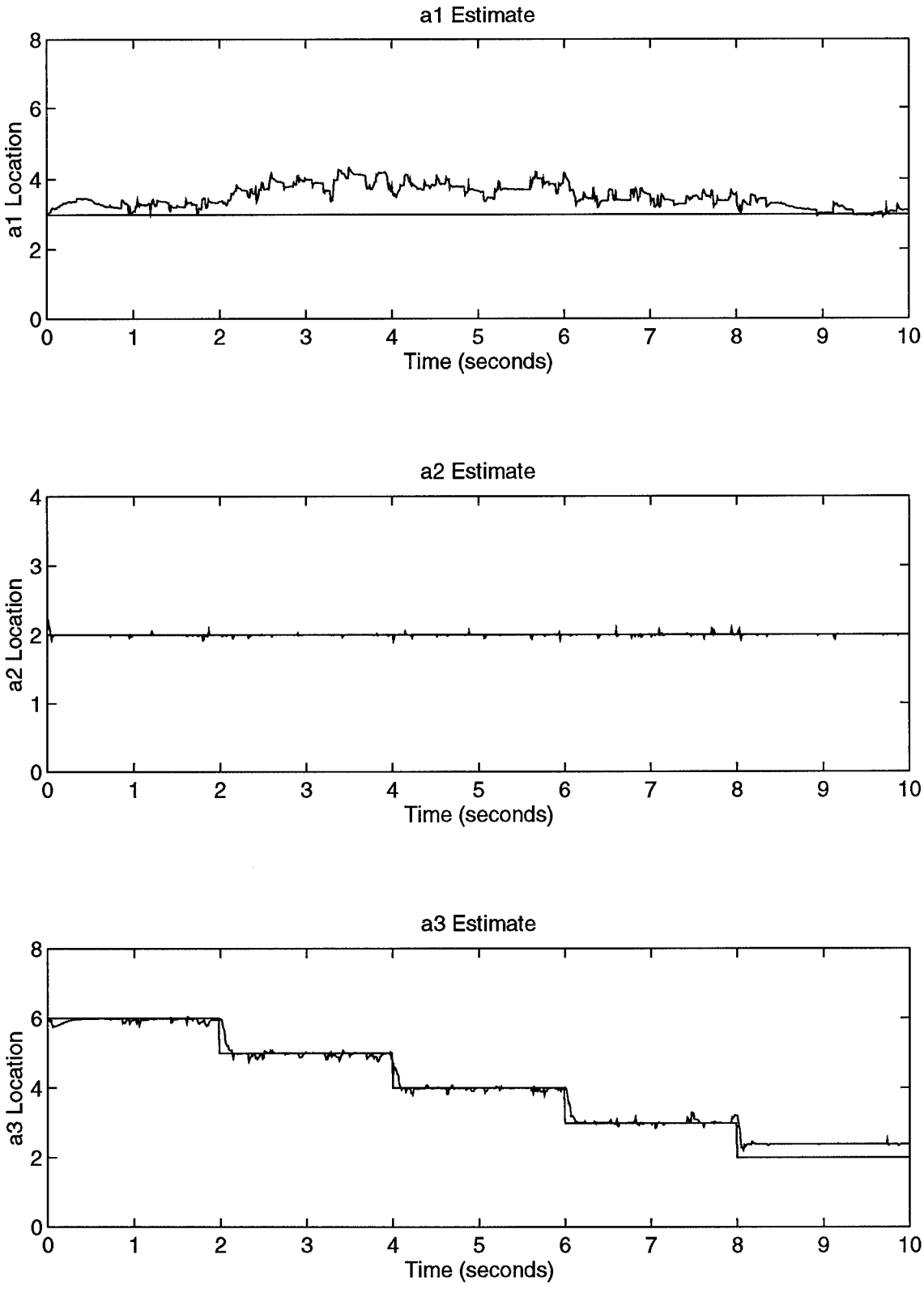
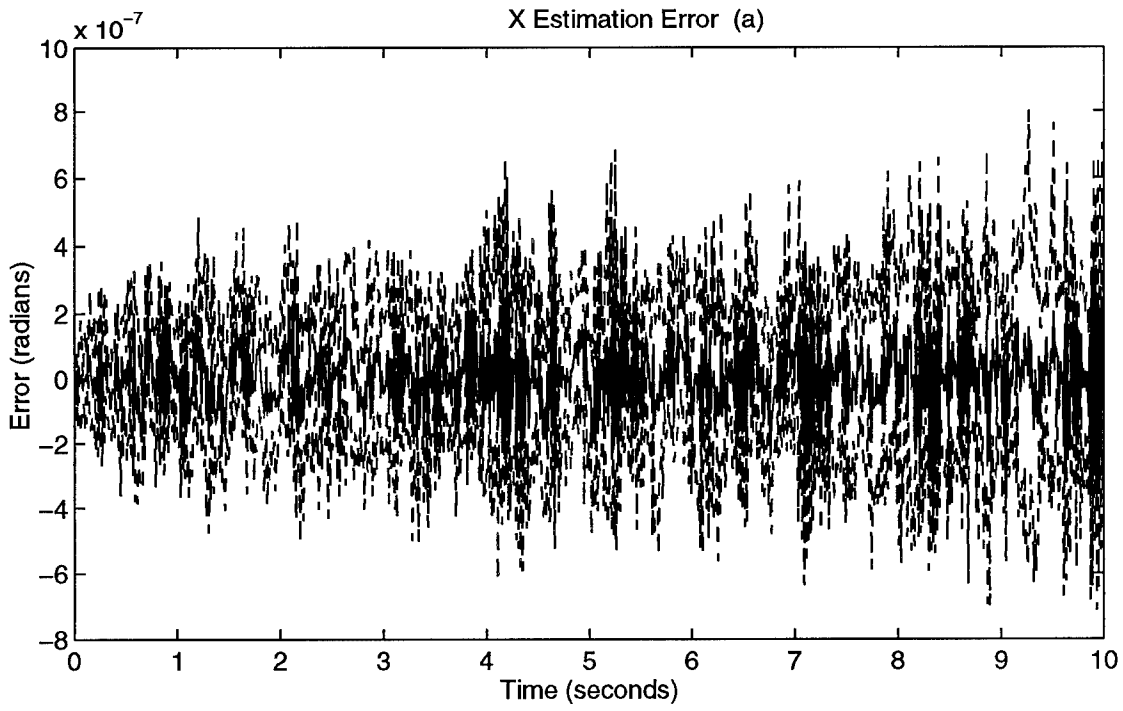


Figure A-28. a_3 Moving Down Using Residual Monitoring



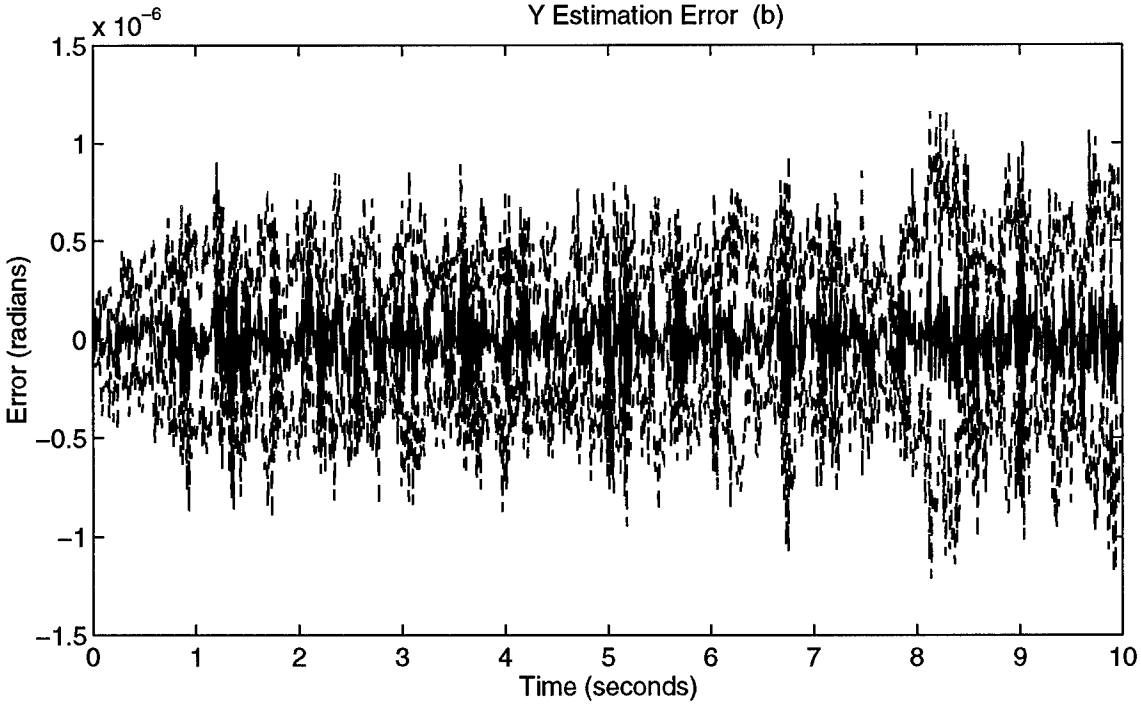


Figure A-29. a_3 Moving Down Estimation Errors

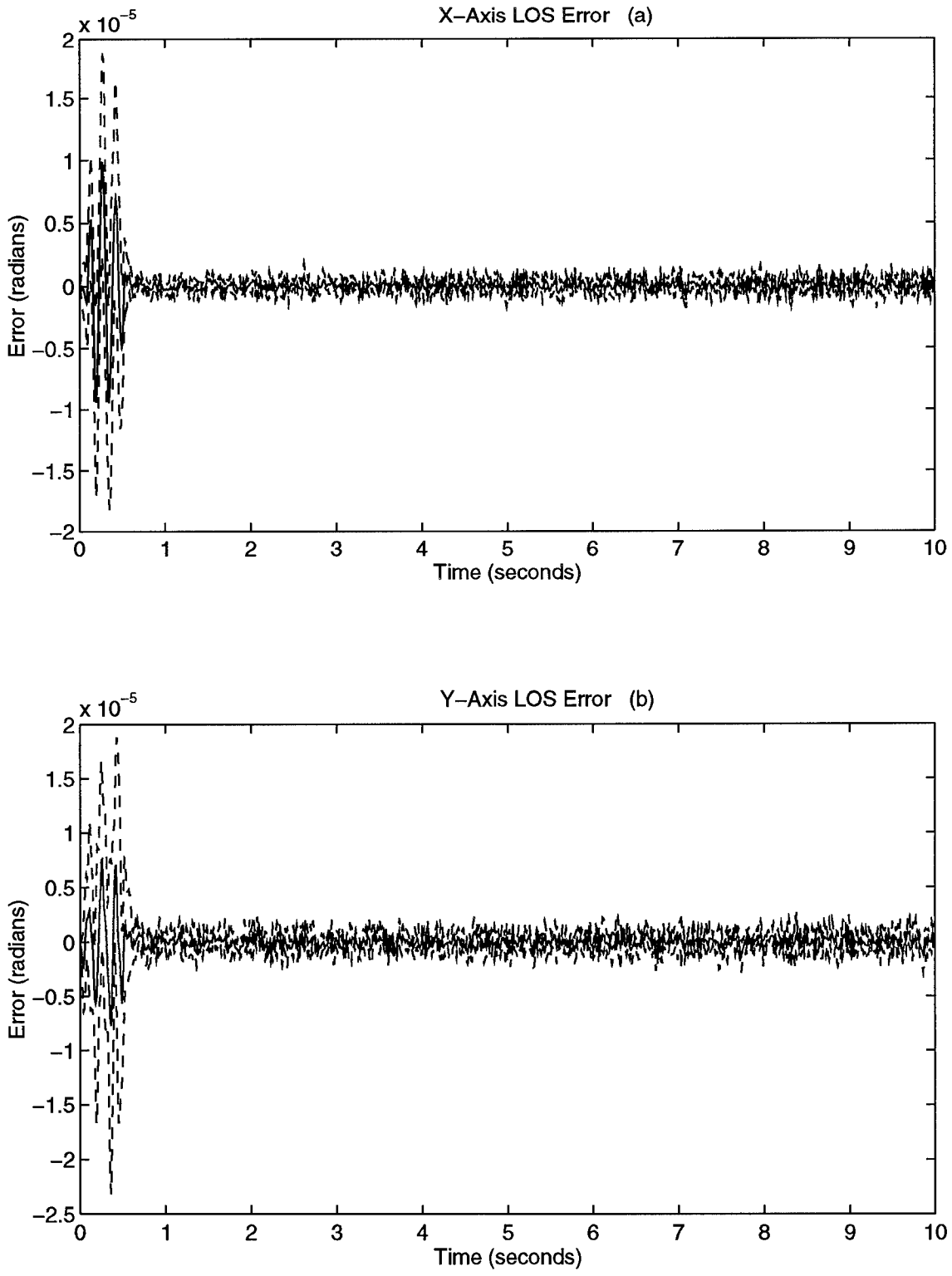


Figure A-30. a_3 Moving Down LOS Errors

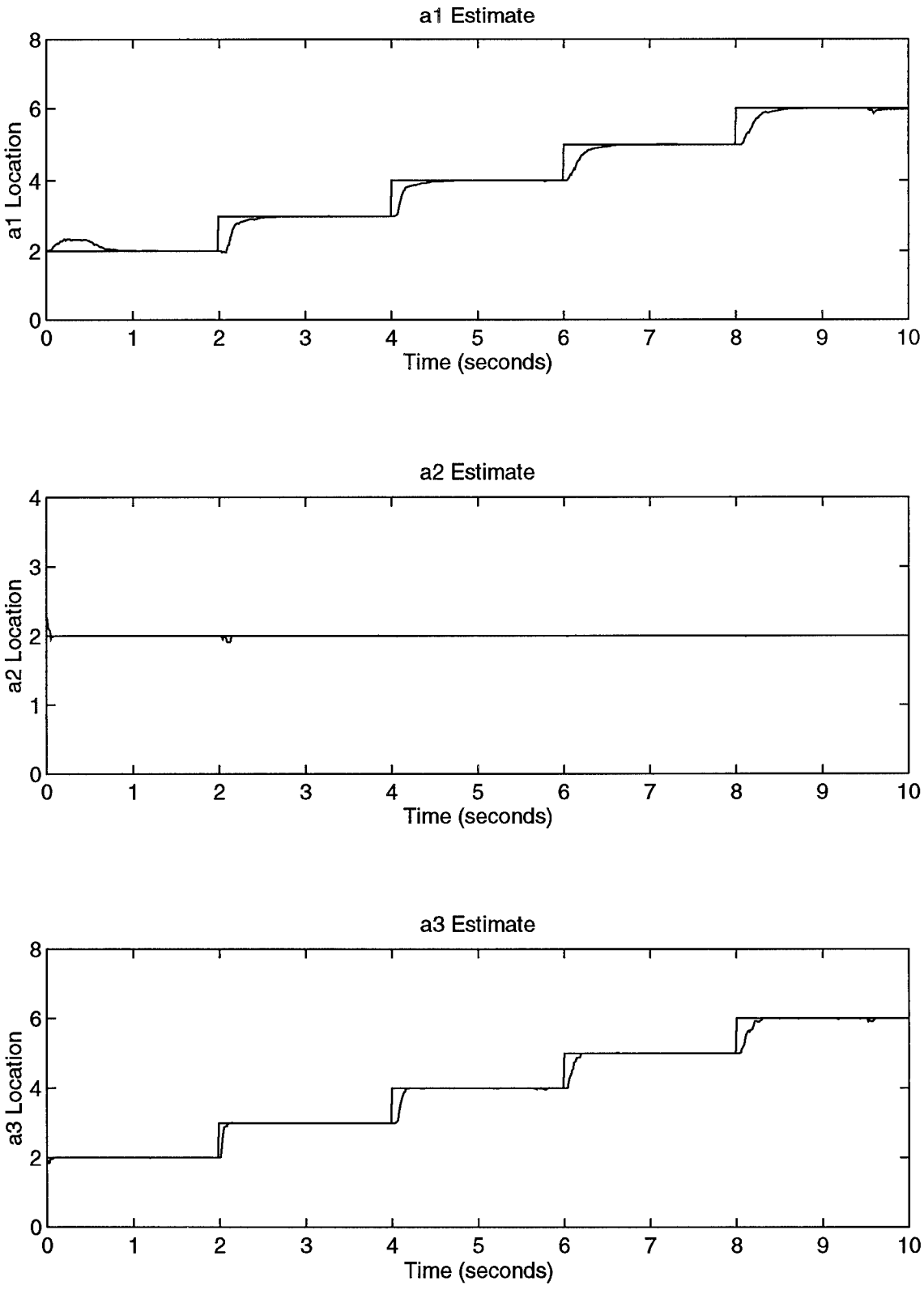


Figure A-31. a_1 and a_3 Moving Up Using Parameter Position Monitoring

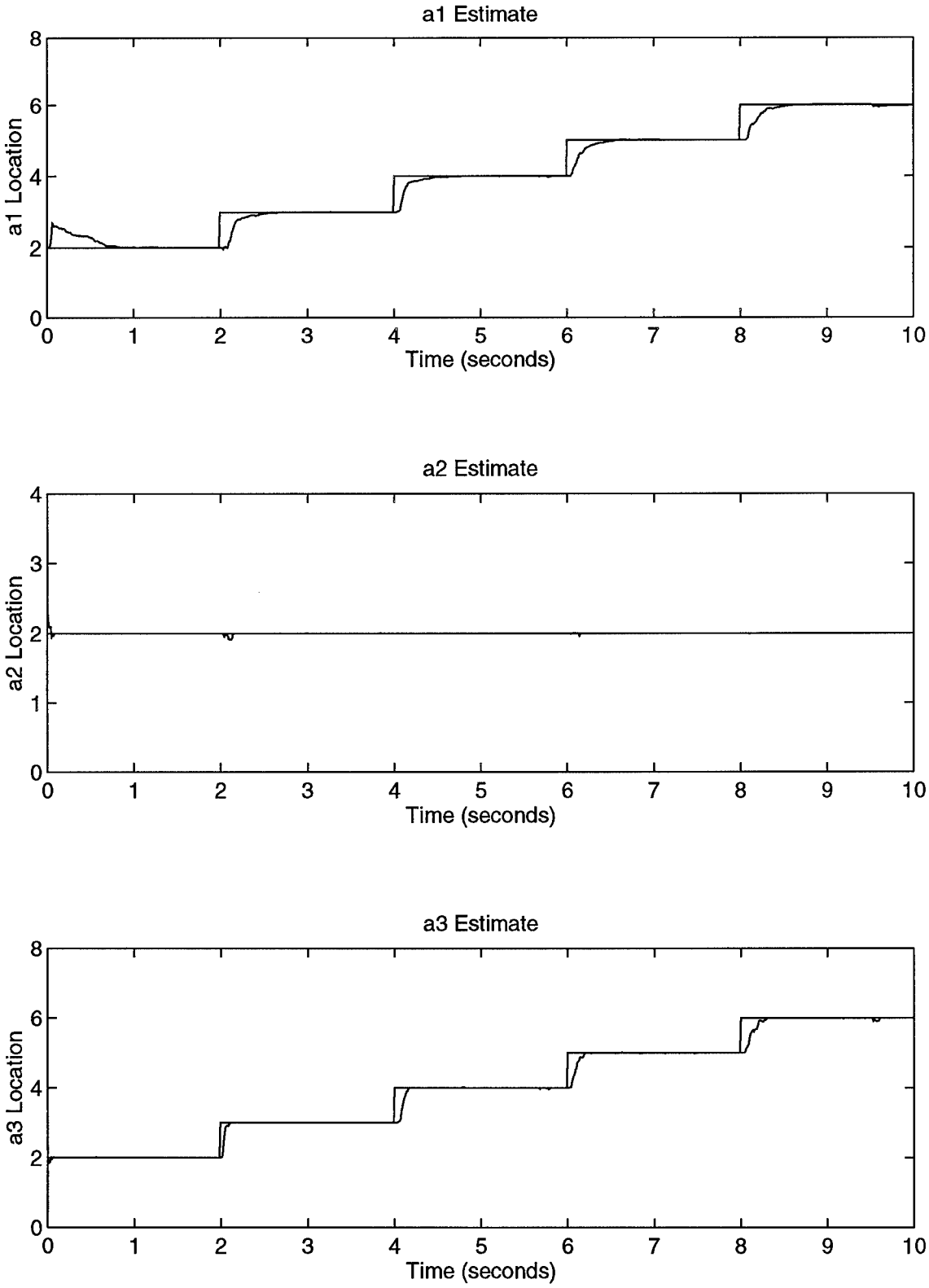


Figure A-32. a_1 and a_3 Moving Up Using Probability Monitoring

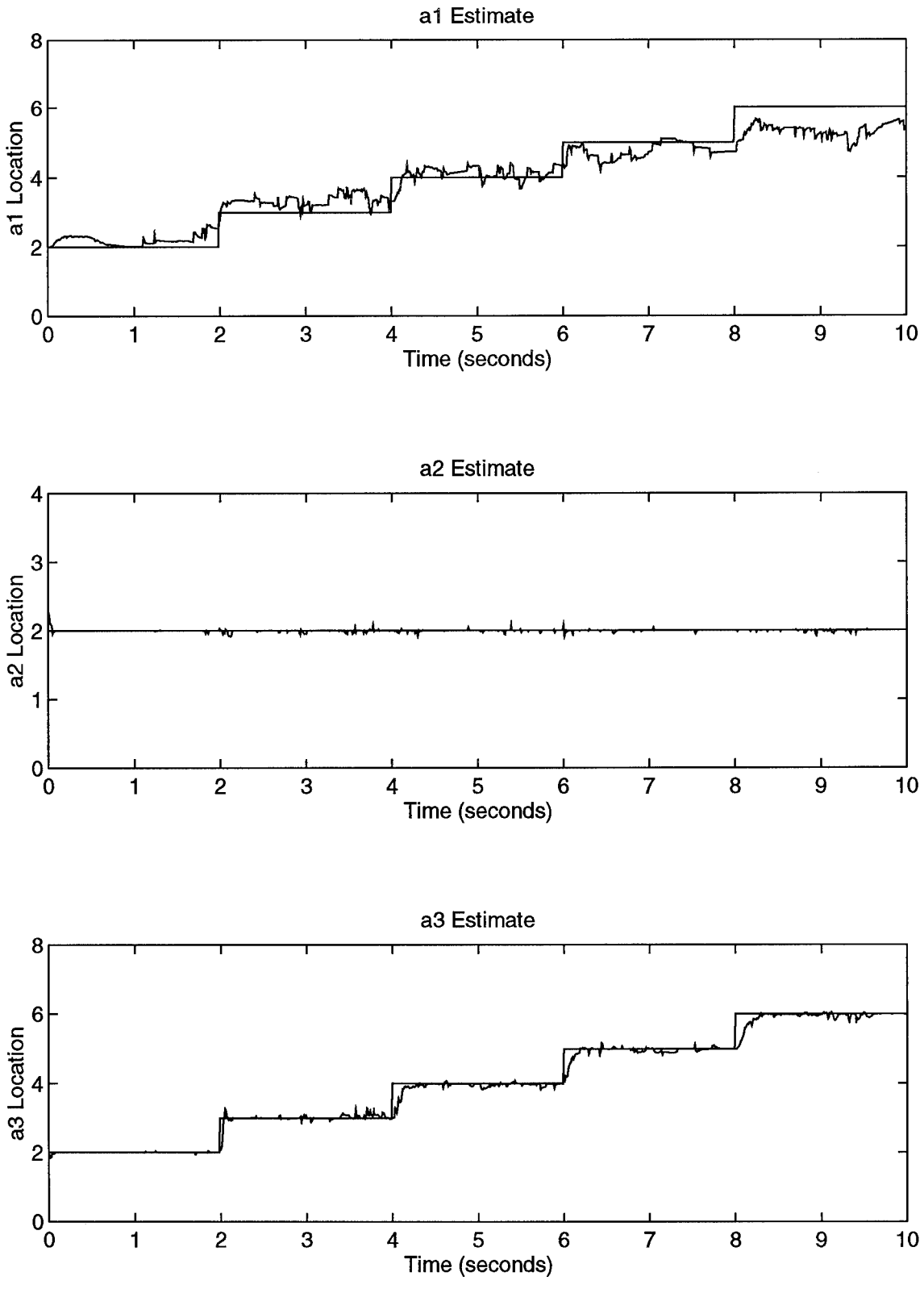


Figure A-33. a_1 and a_3 Moving Up Using Residual Monitoring

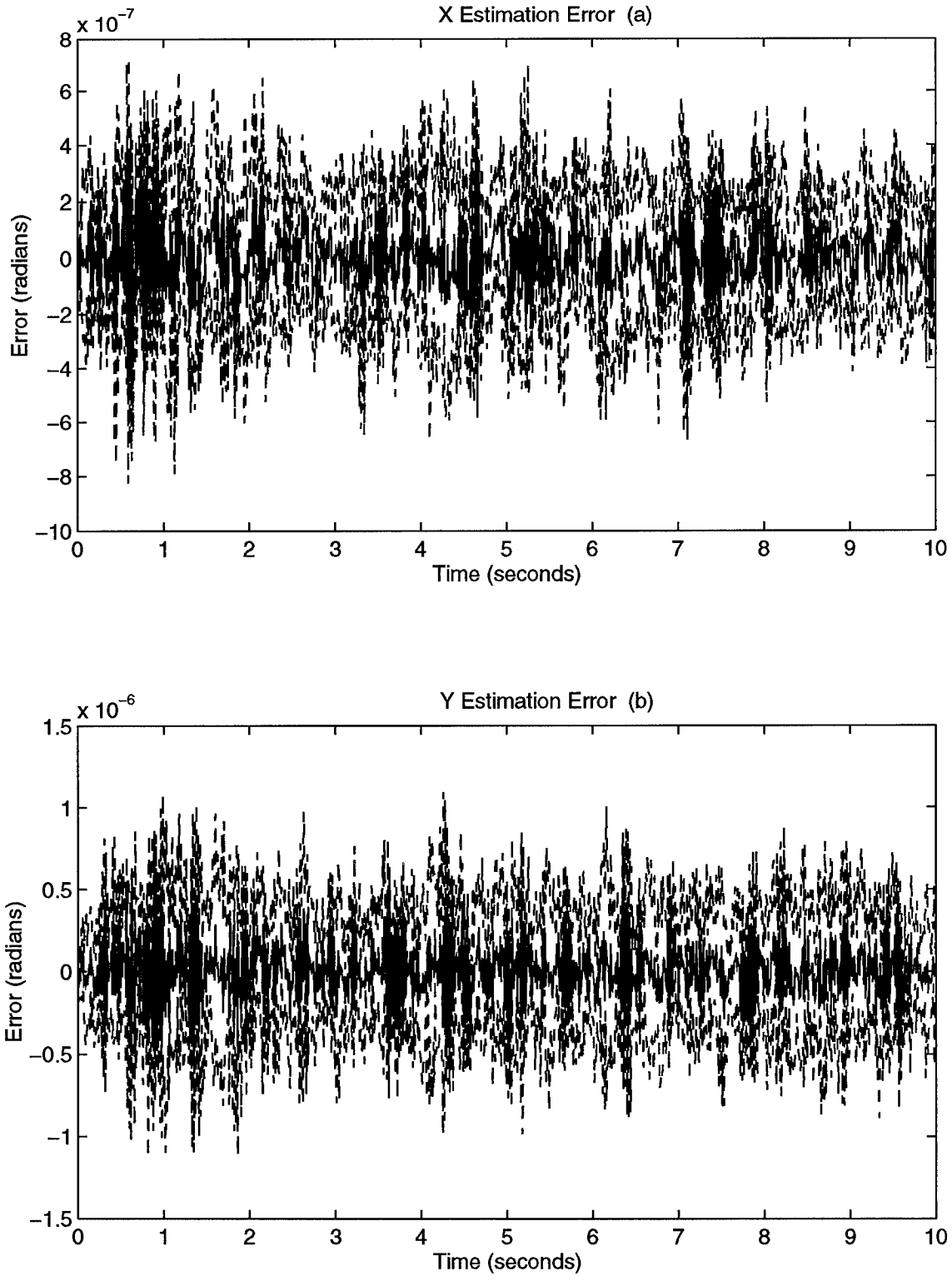
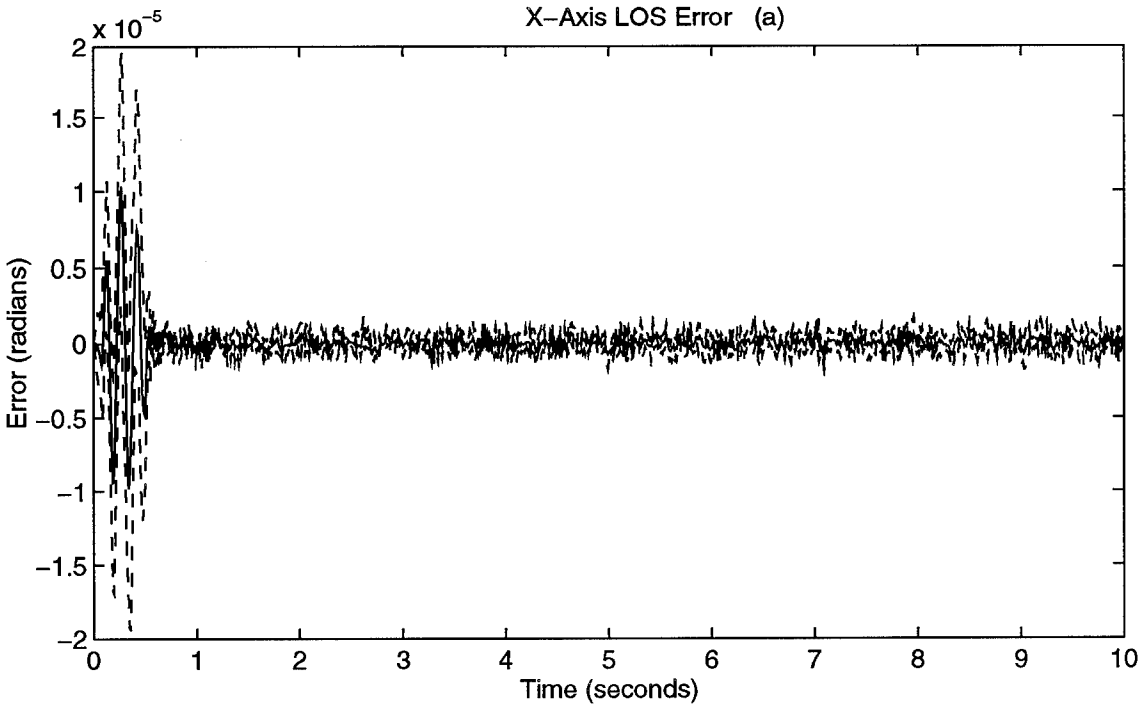


Figure A-34. a_1 and a_3 Moving Up Estimation Errors



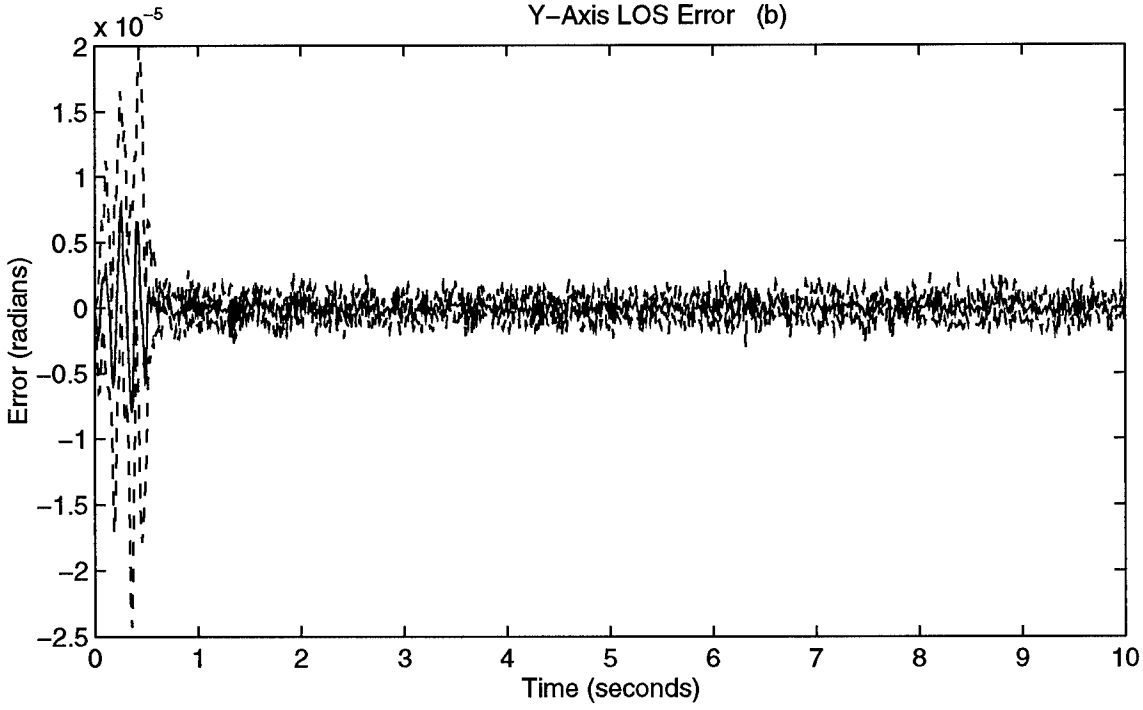


Figure A-35. a_1 and a_3 Moving Up LOS Errors

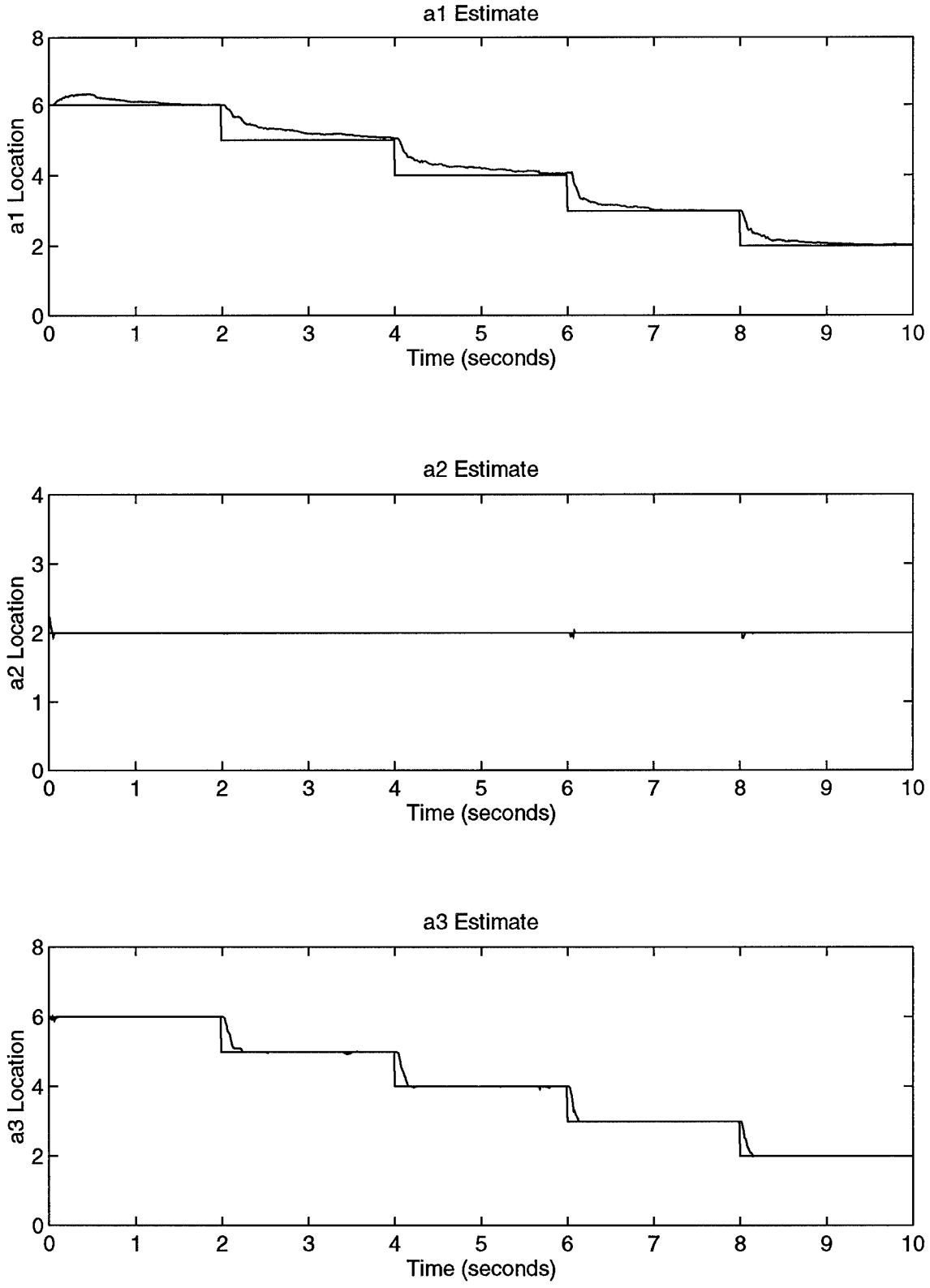


Figure A-36. a_1 and a_3 Moving Down Using Parameter Position Monitoring

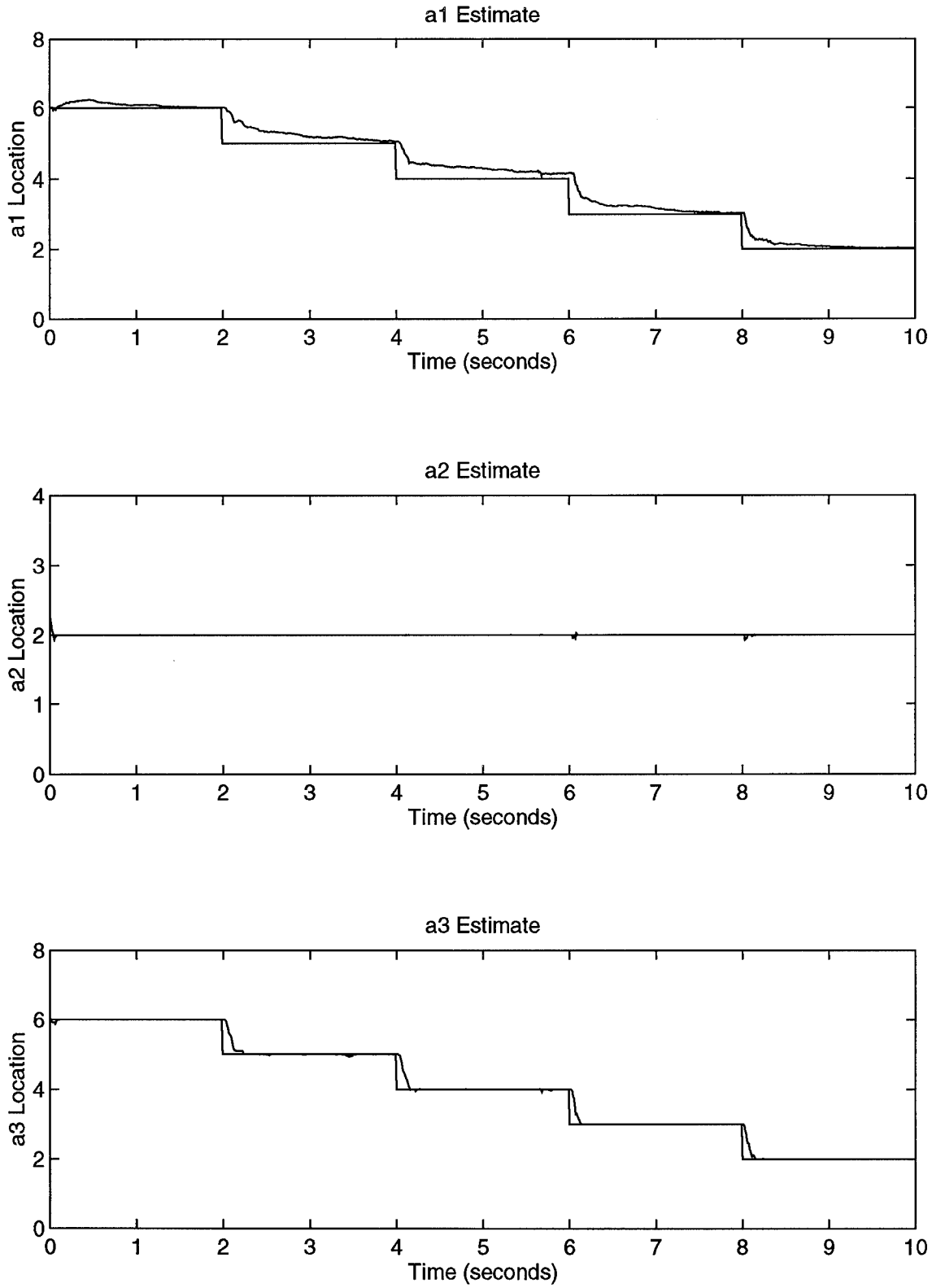


Figure A-37. a_1 and a_3 Moving Down Using Probability Monitoring

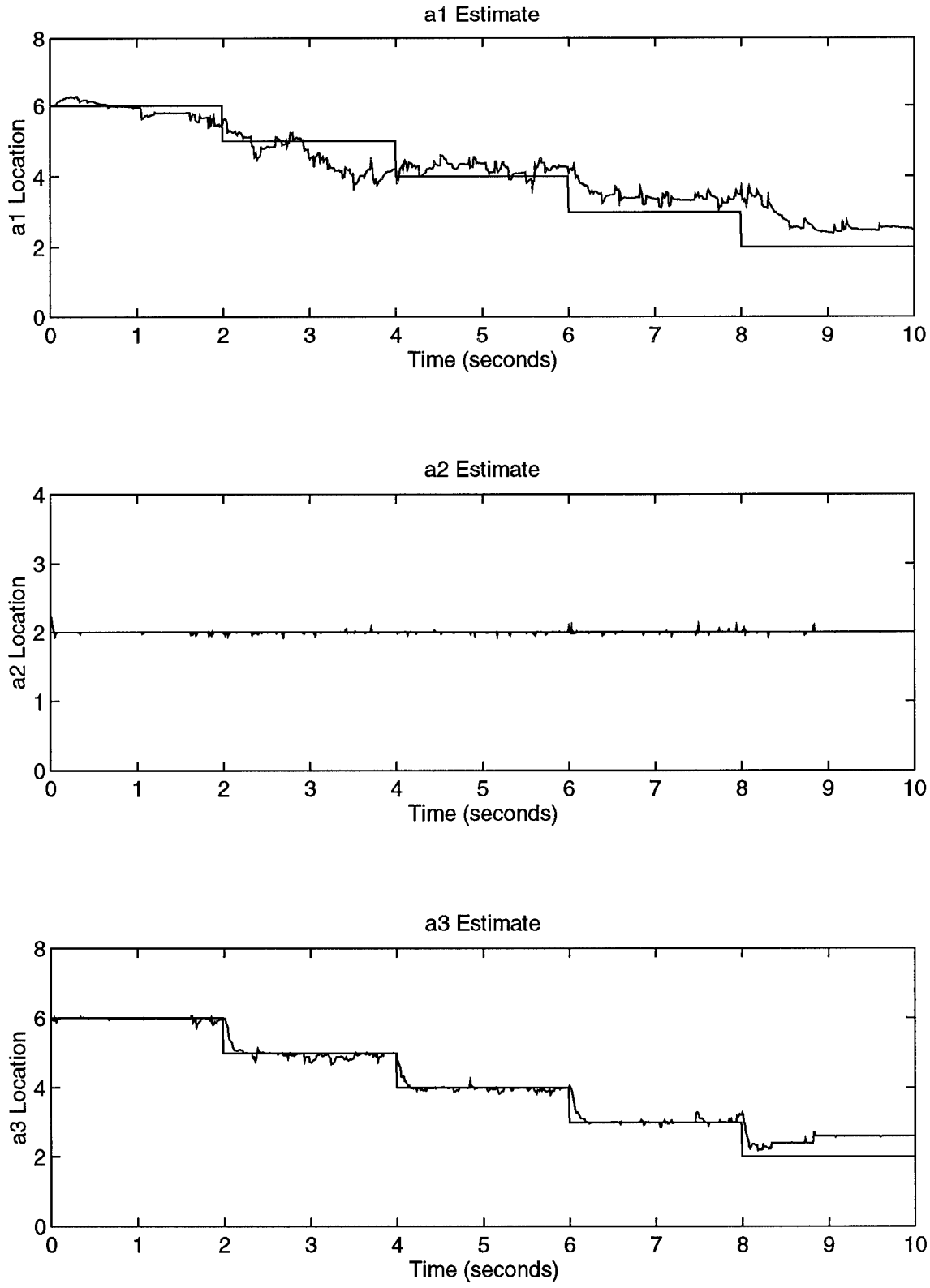
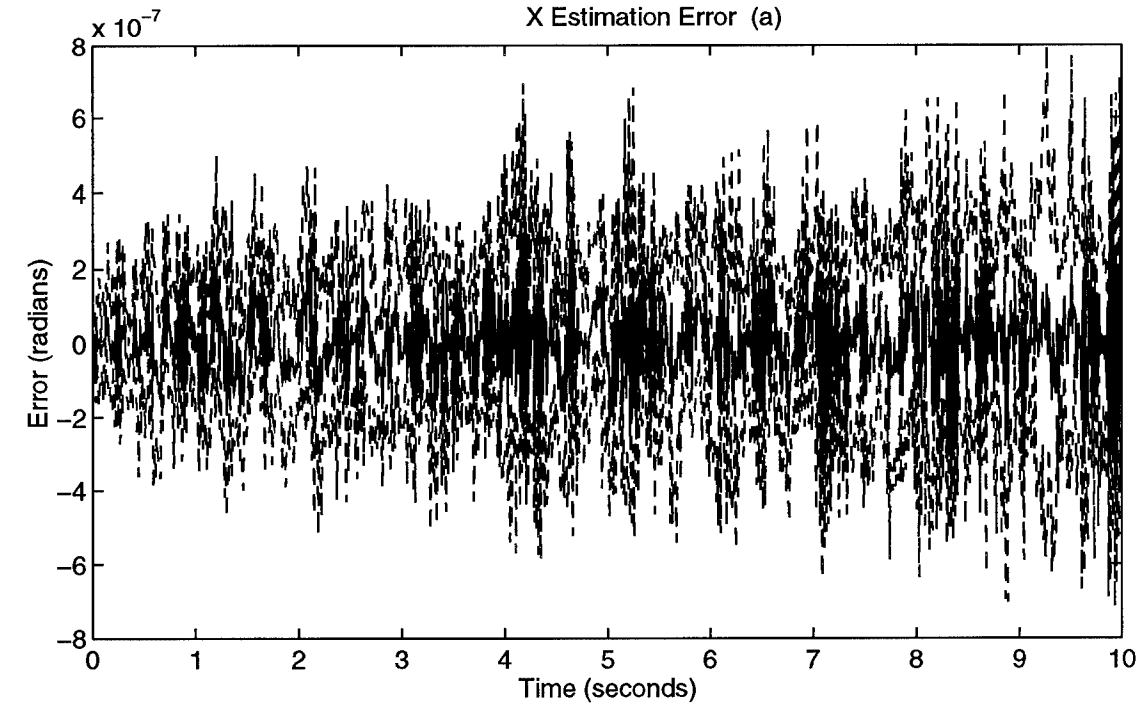


Figure A-38. a_1 and a_3 Moving Down Using Residual Monitoring



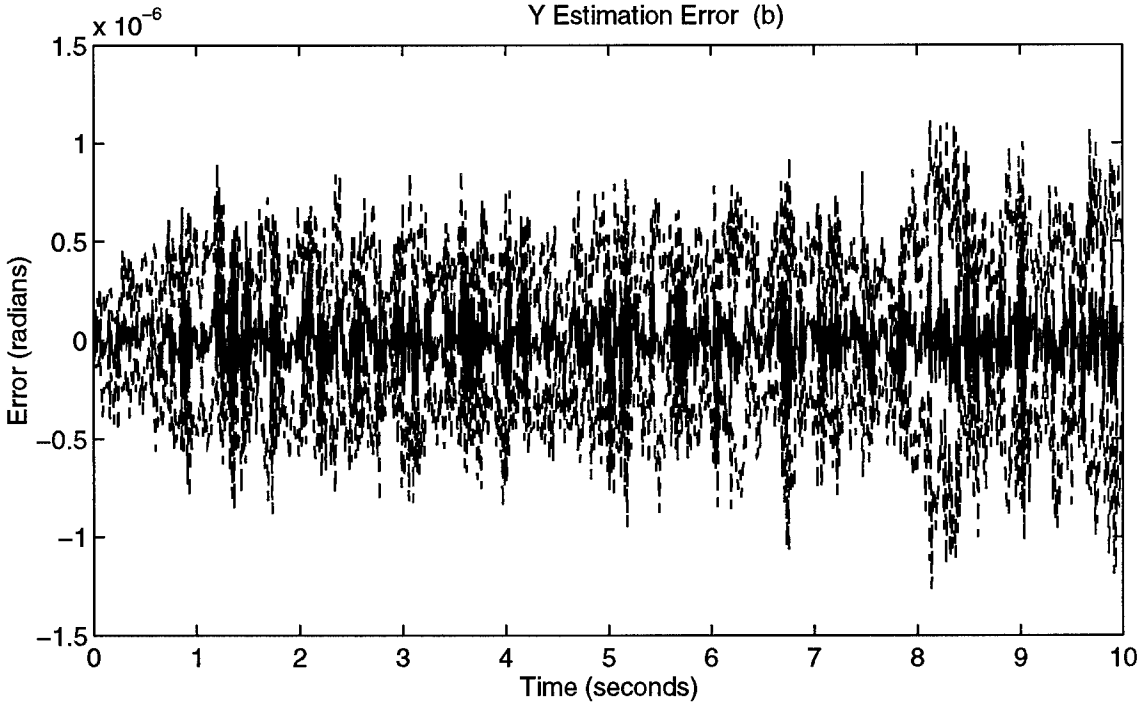


Figure A-39. a_1 and a_3 Moving Down Estimation Errors

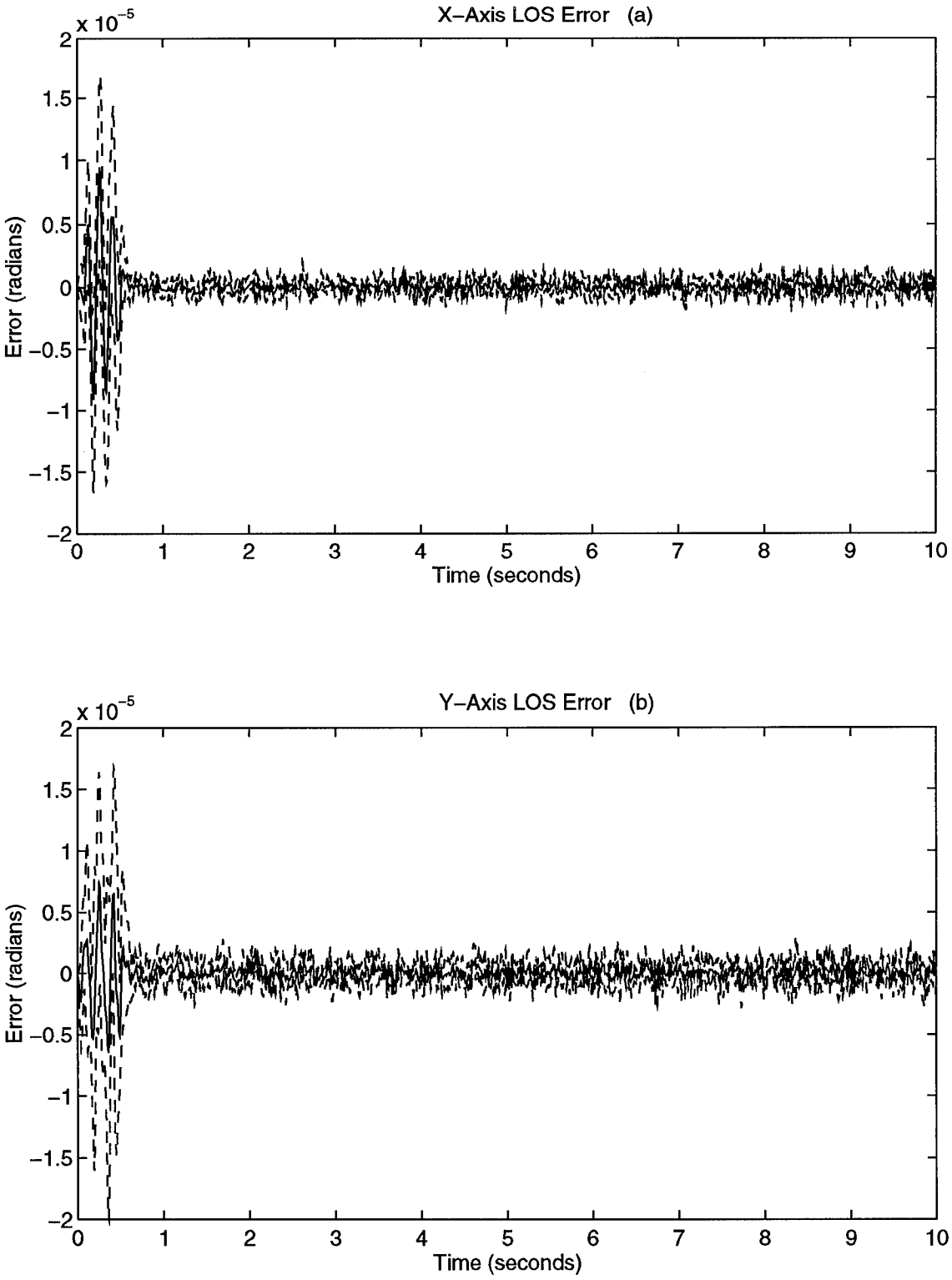


Figure A-40. a_1 and a_3 Moving Down LOS Errors

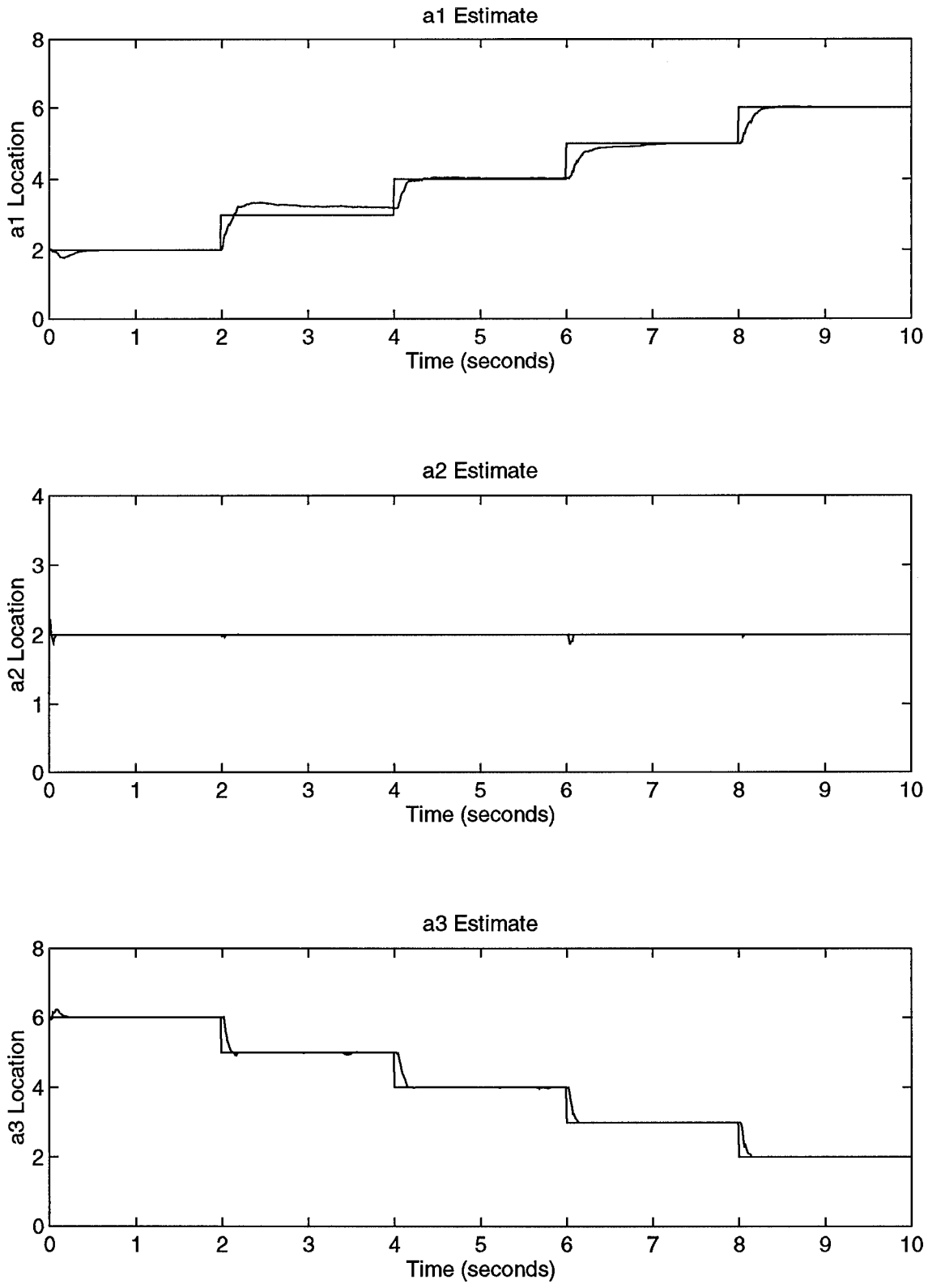


Figure A-41. a_1 Moving Up/a_3 Moving Down Using Parameter Position Monitoring

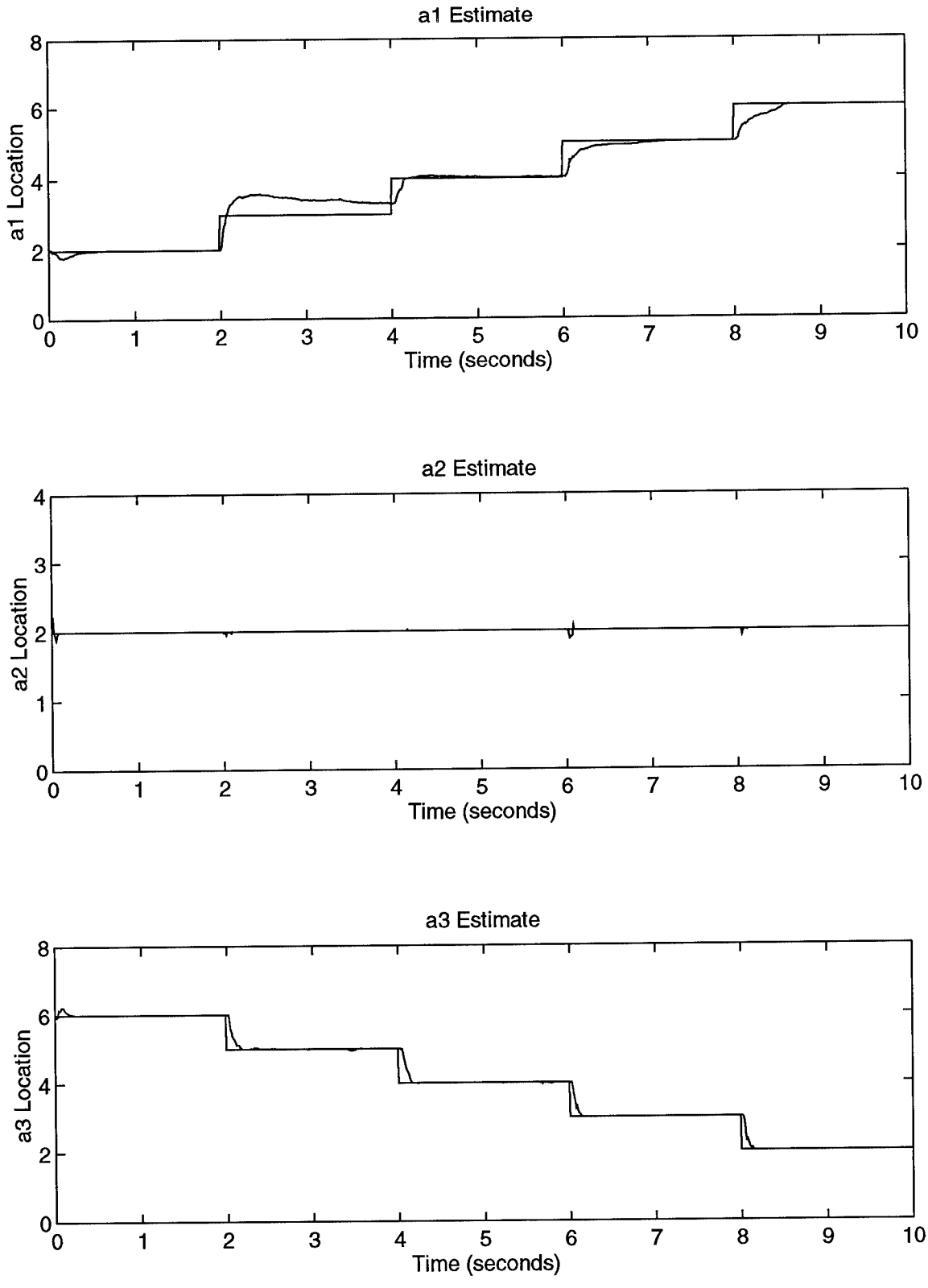


Figure A-42. a_1 Moving Up/a_3 Moving Down Using Probability Monitoring

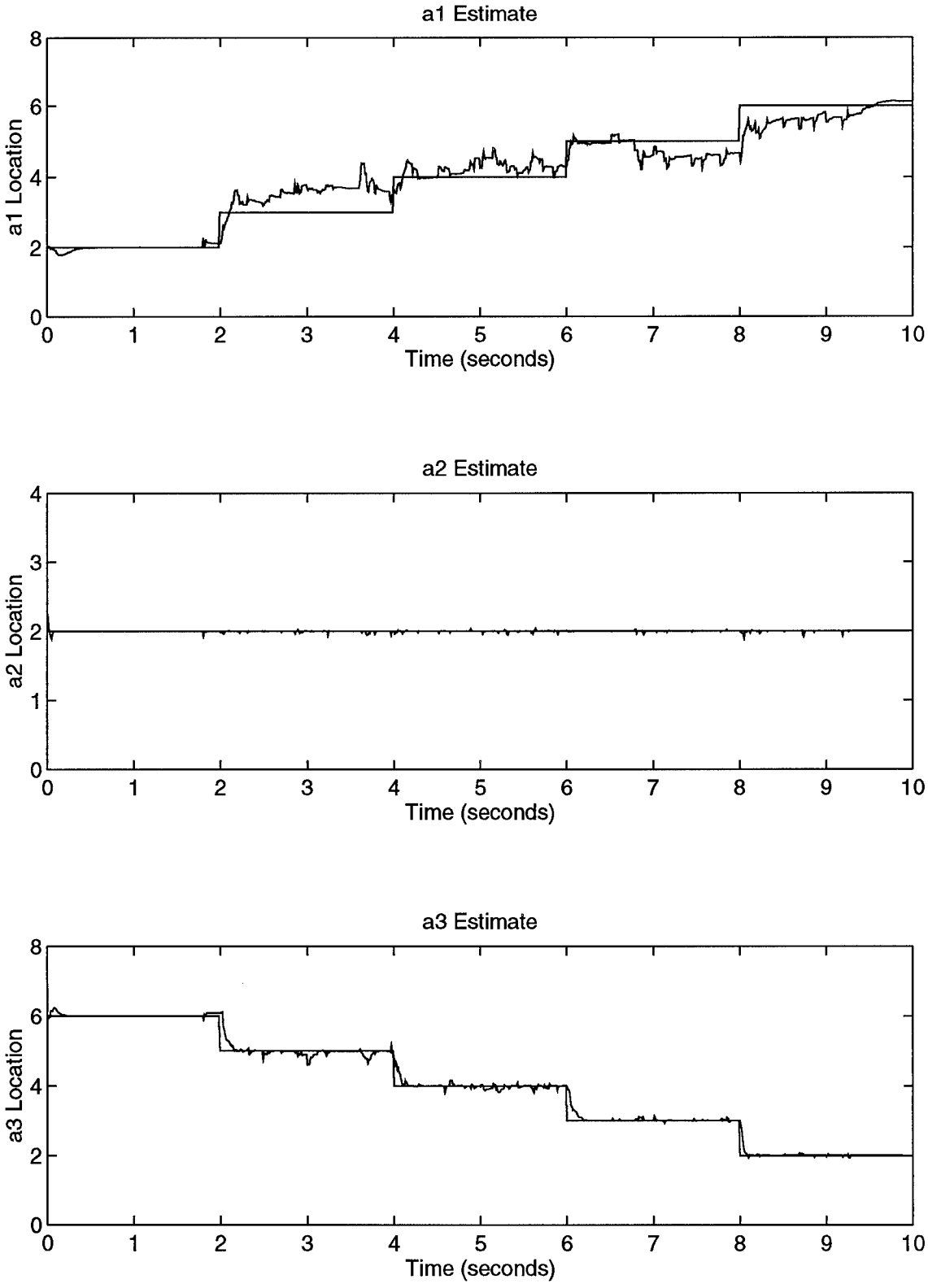


Figure A-43. a_1 Moving Up/a_3 Moving Down Using Residual Monitoring

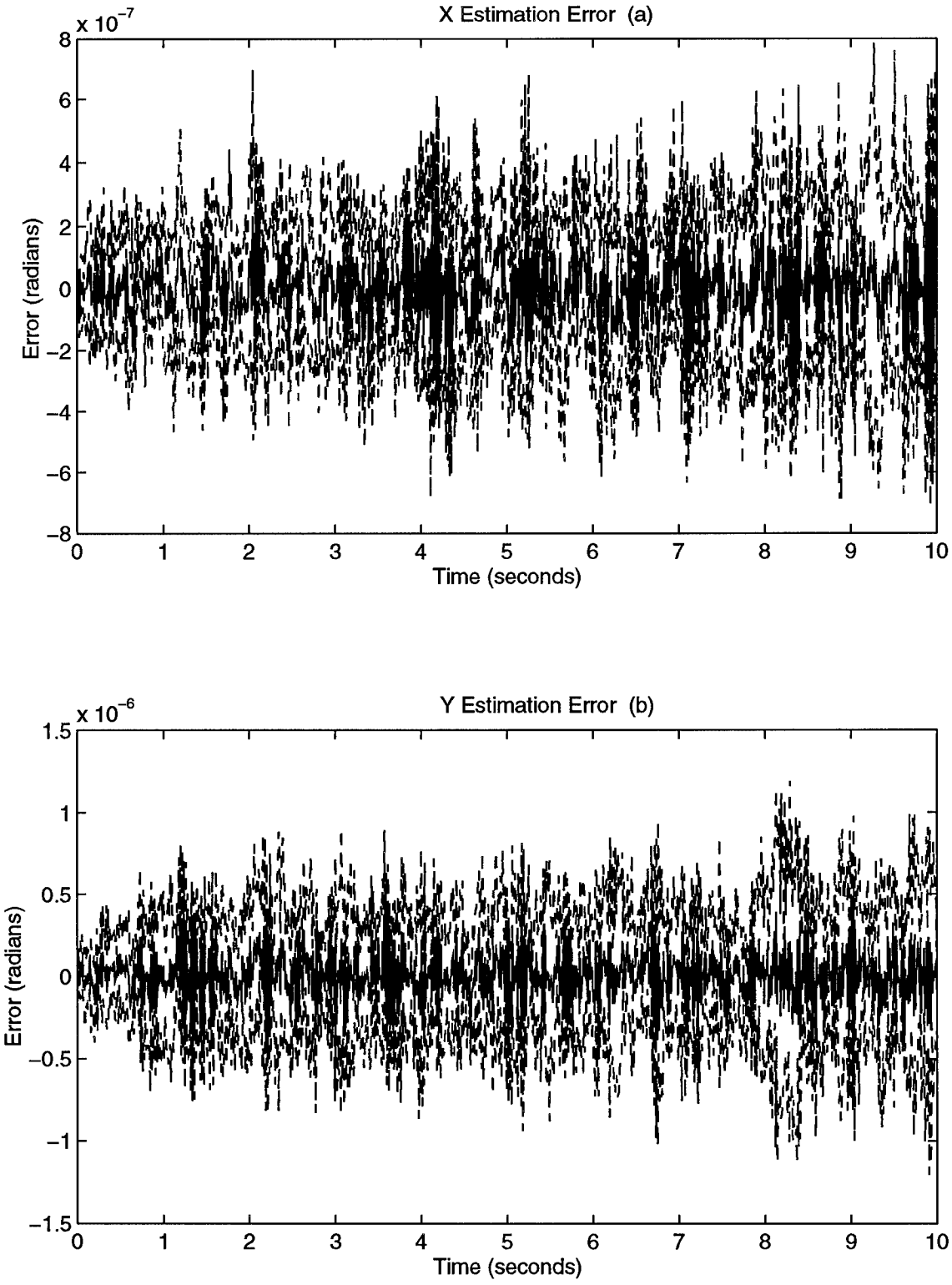
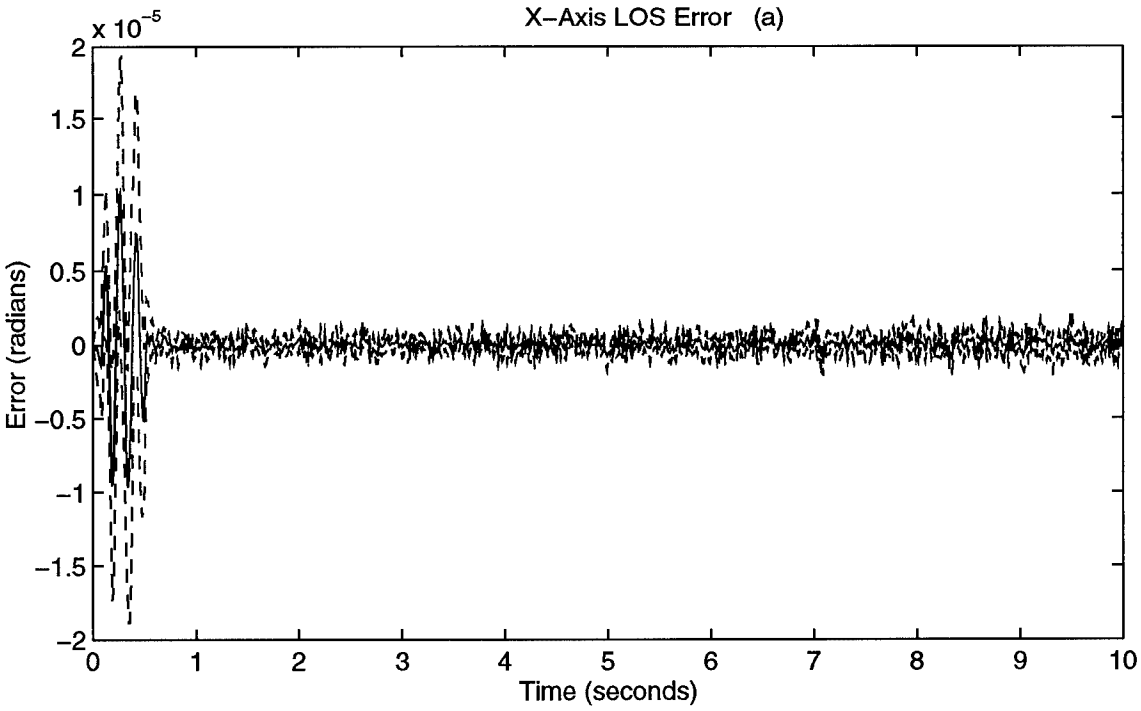


Figure A-44. a_1 Moving Up/a_3 Moving Down Estimation Errors



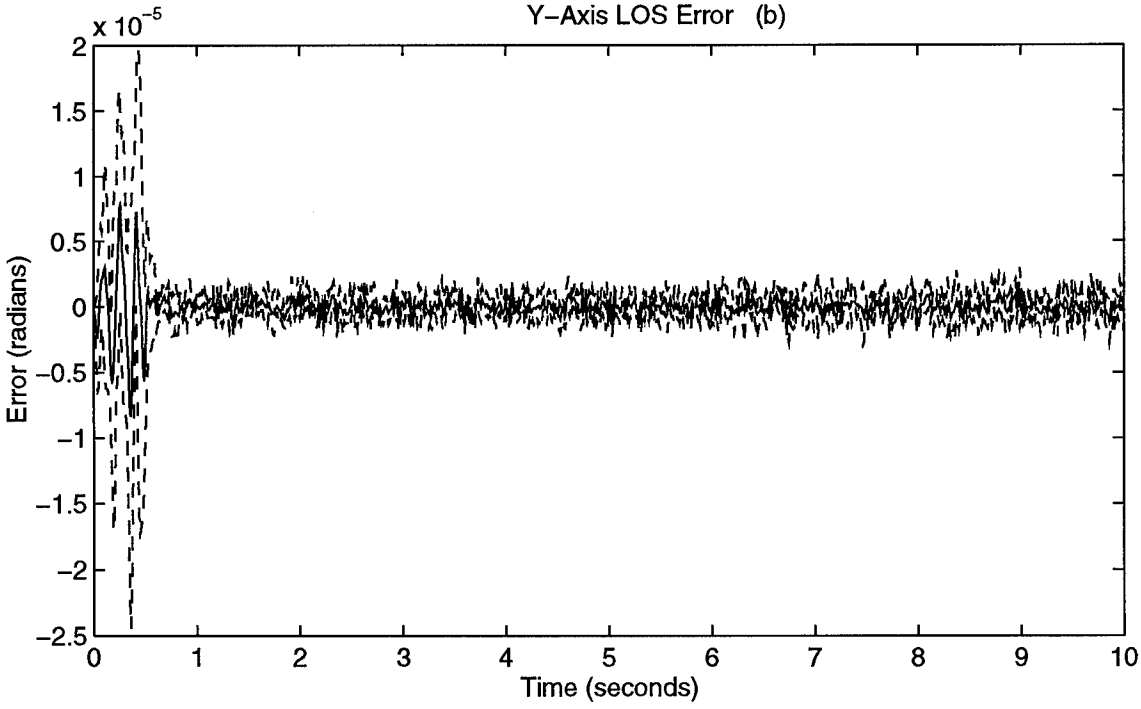


Figure A-45. a_1 Moving Up/a_3 Moving Down LOS Errors

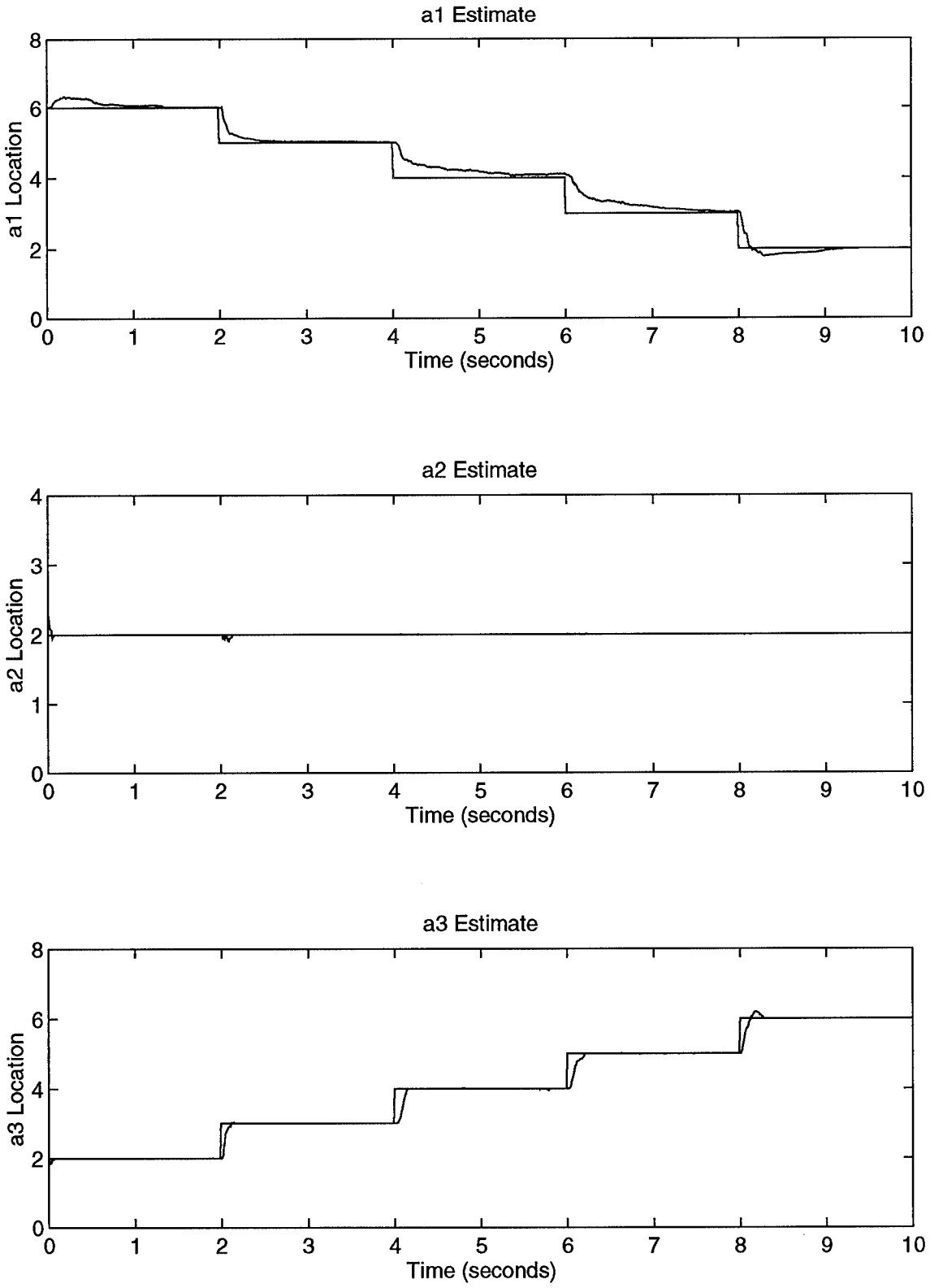


Figure A-46. a_1 Moving Down/a_3 Moving Up Using Parameter Position Monitoring

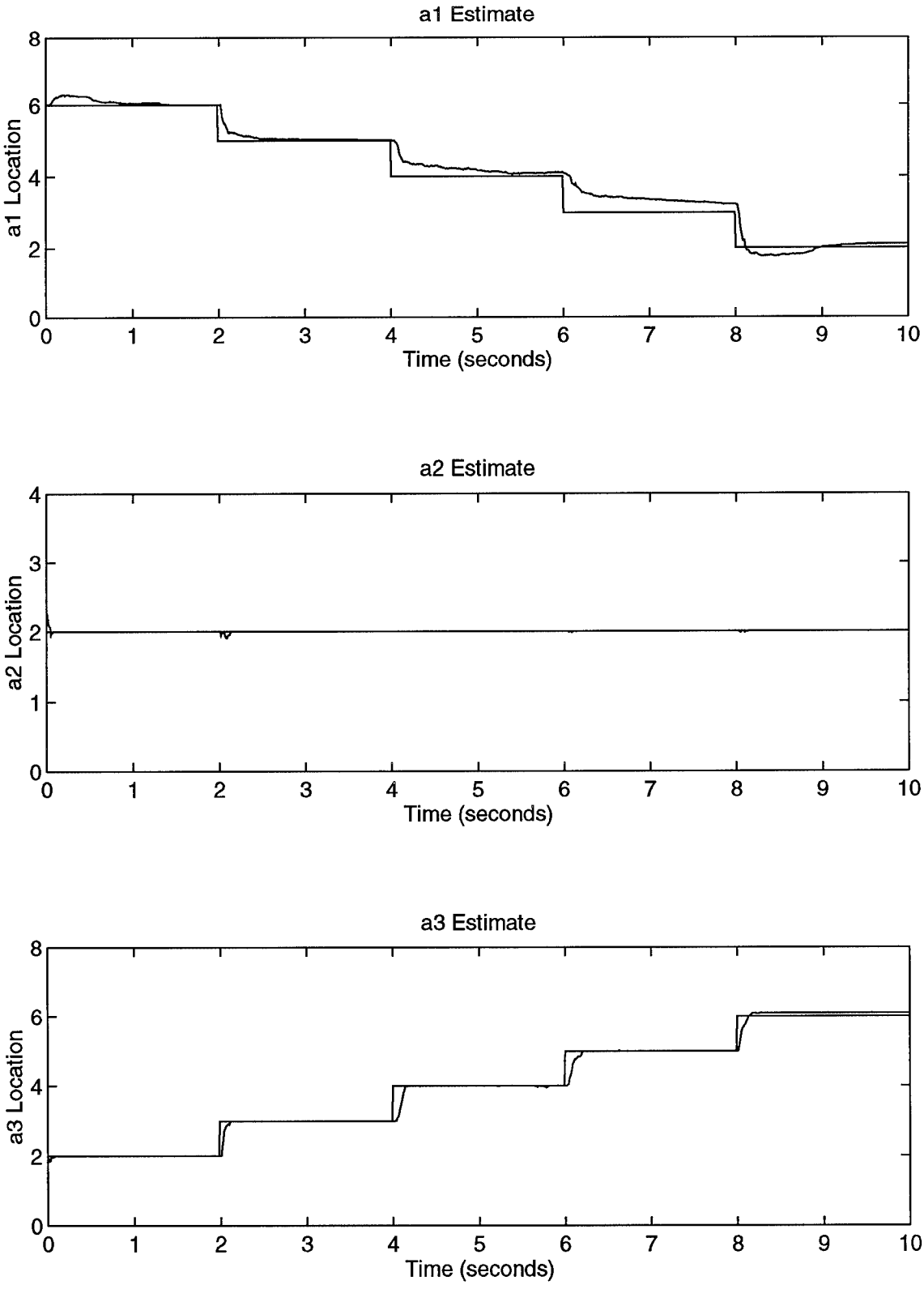


Figure A-47. a_1 Moving Down/a_3 Moving Up Using Probability Monitoring

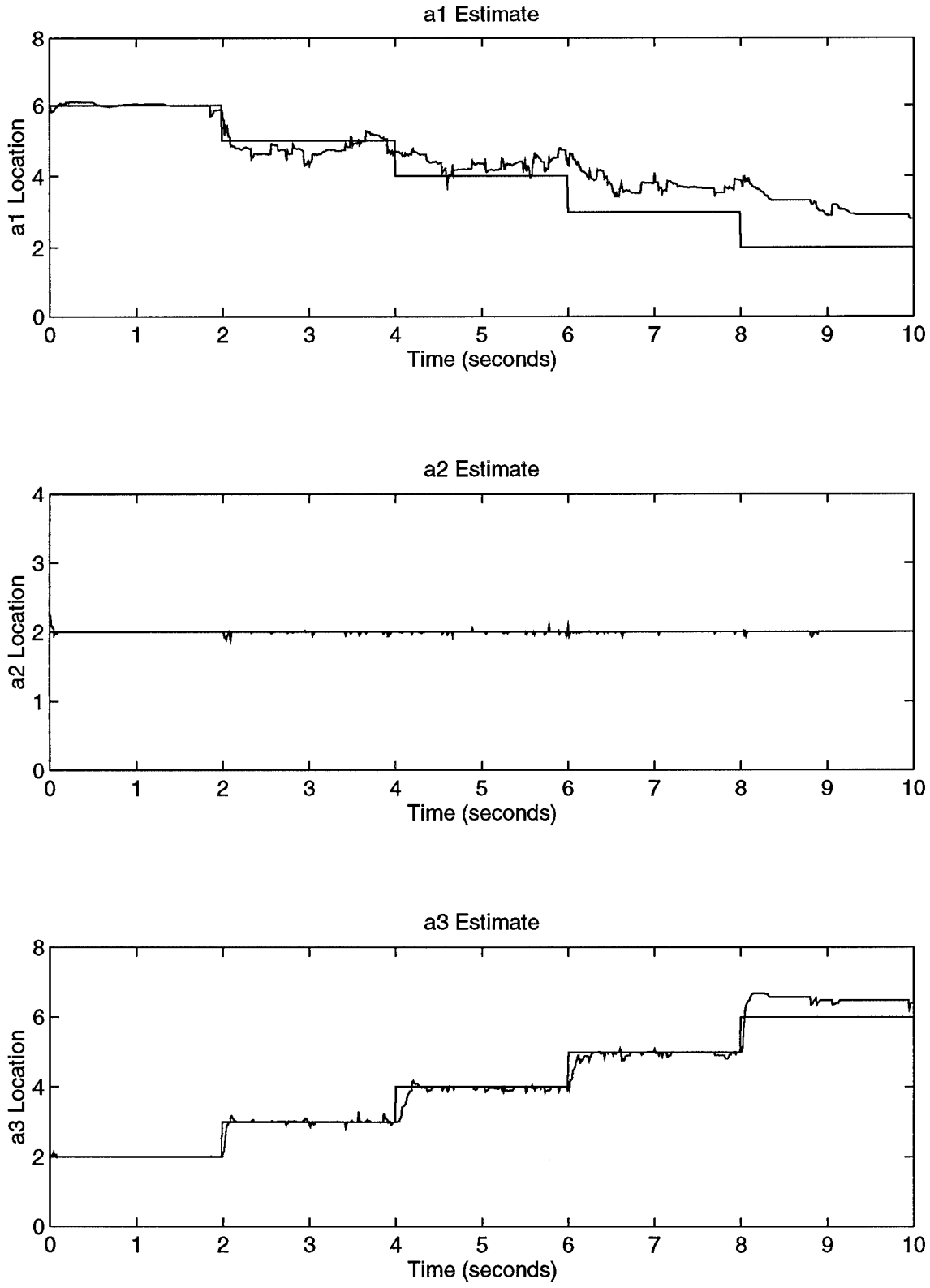


Figure A-48. a_1 Moving Down/a_3 Moving Up Using Residual Monitoring

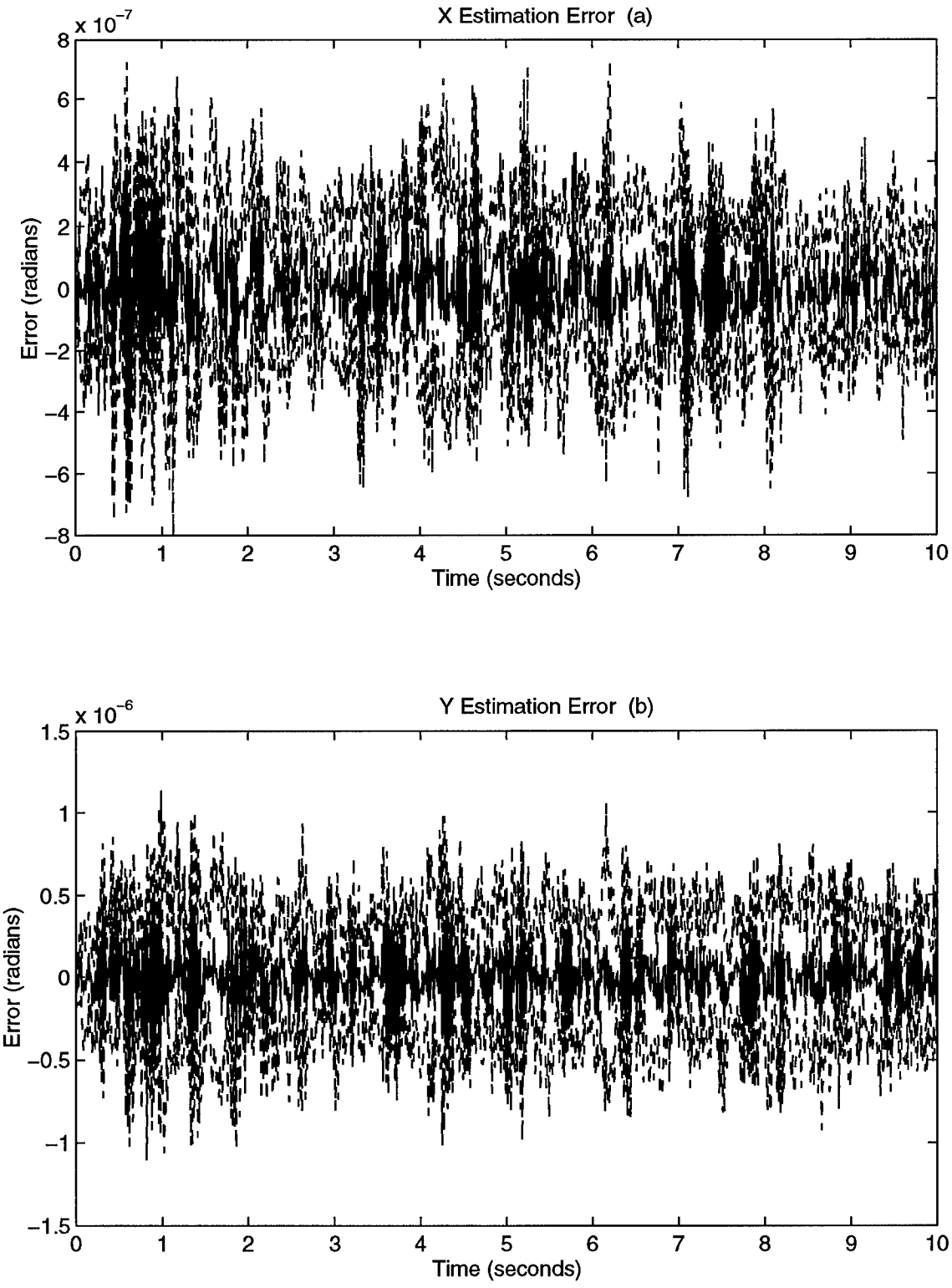
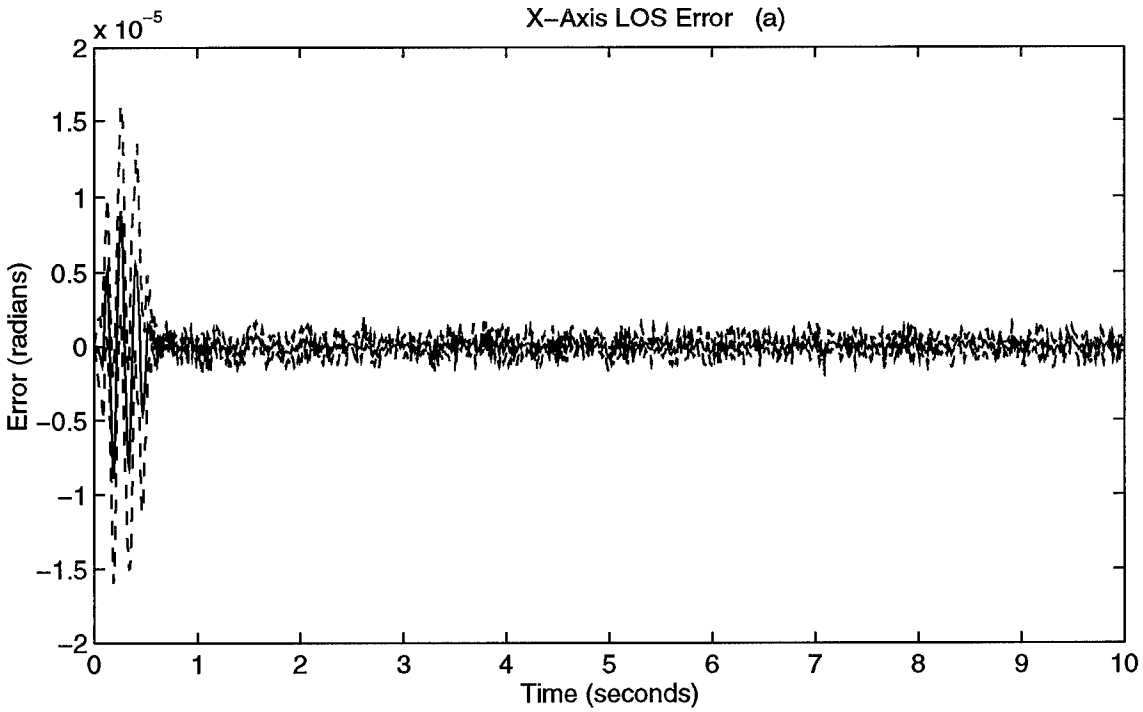


Figure A-49. a_1 Moving Down/a_3 Moving Up Estimation Errors



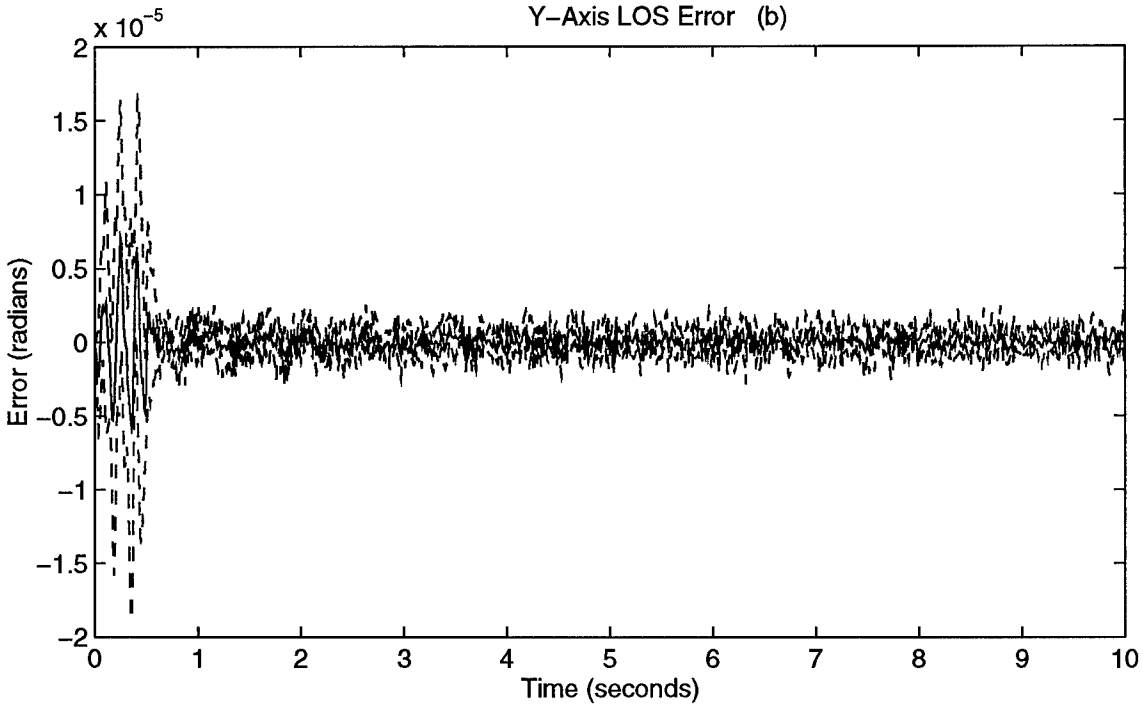


Figure A-50. a_1 Moving Down/a_3 Moving Up LOS Errors

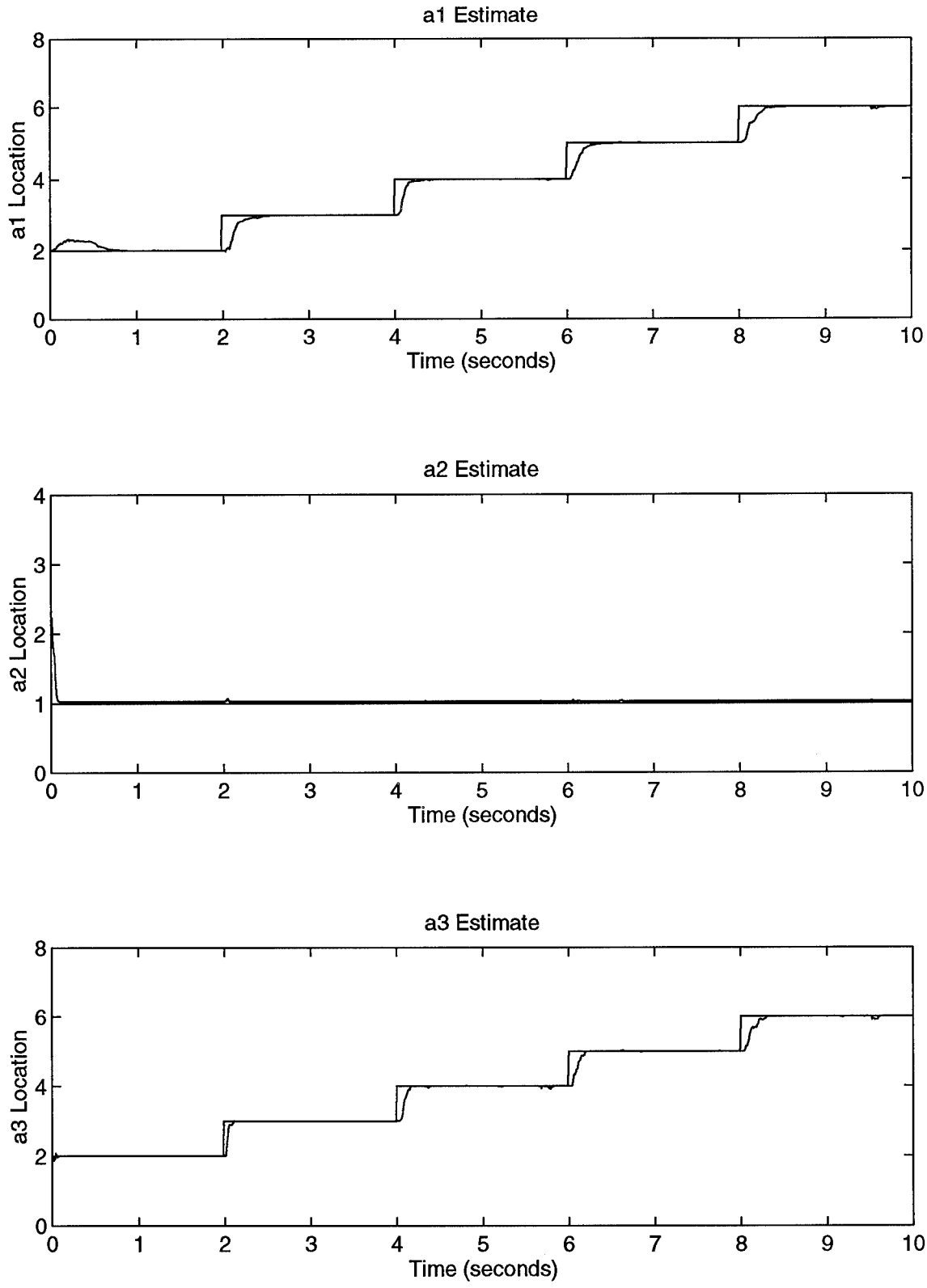


Figure A-51. a_1 and a_3 Moving Up ($a_2 = 1$) Using Parameter Position Monitoring

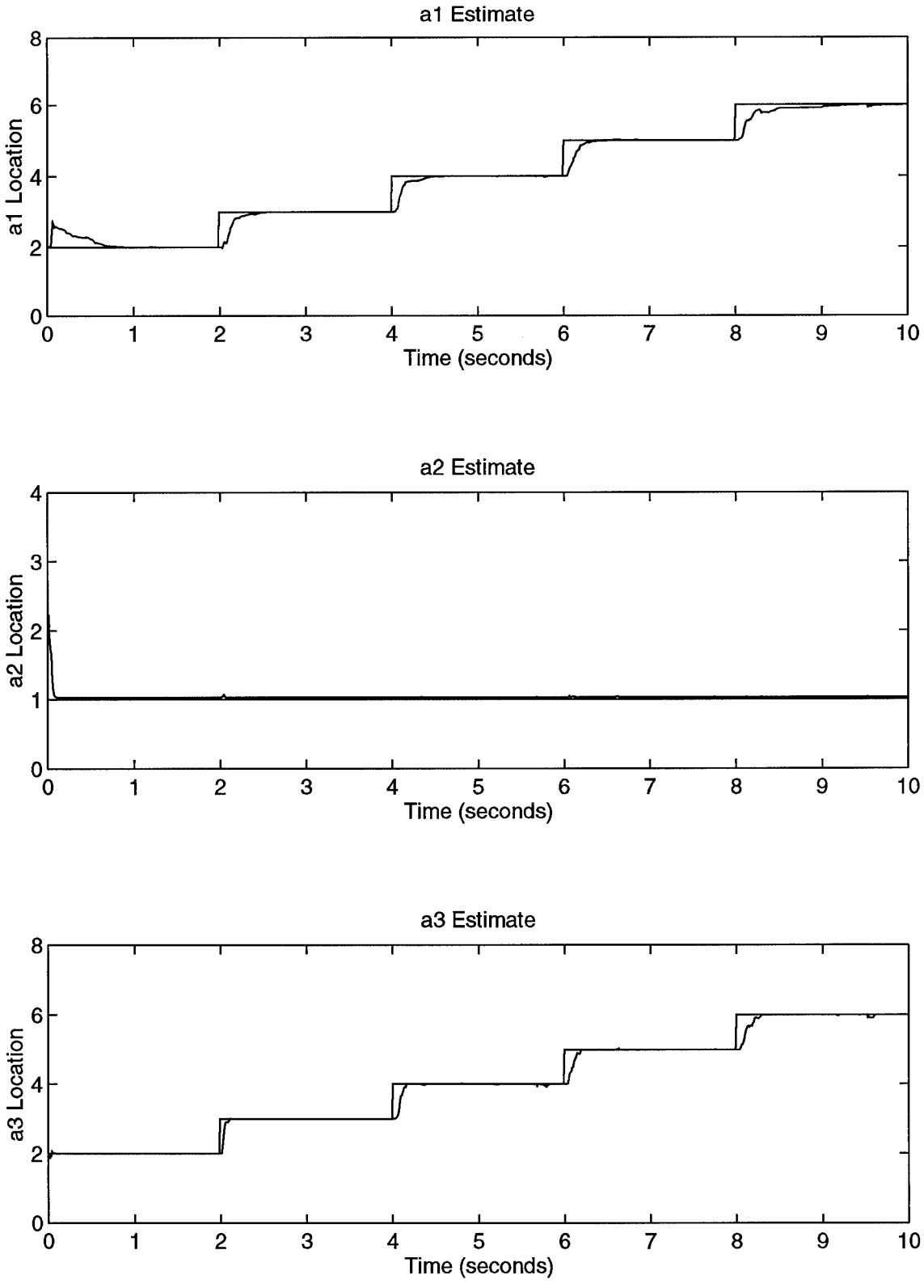


Figure A-52. a_1 and a_3 Moving Up ($a_2 = 1$) Using Probability Monitoring

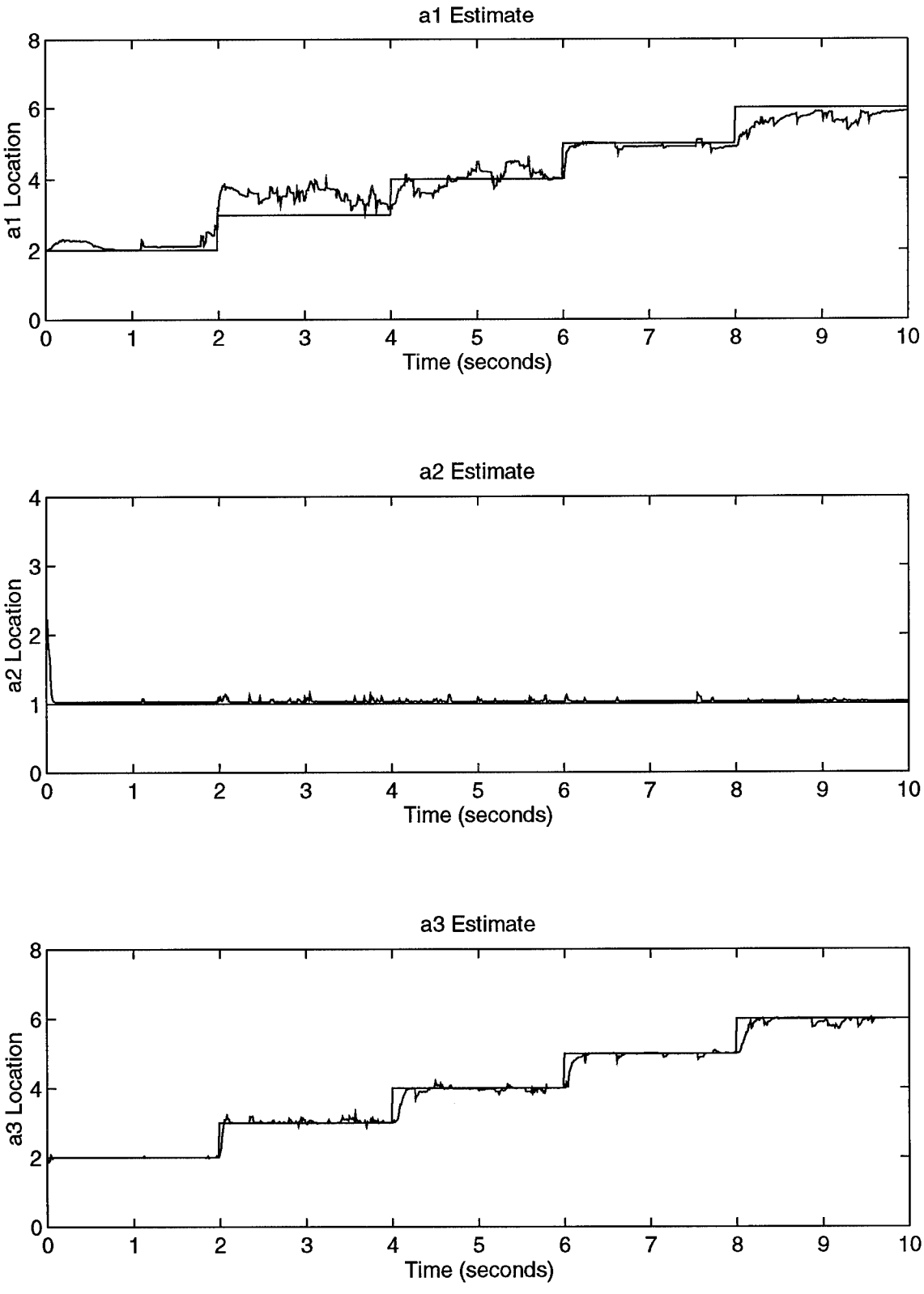
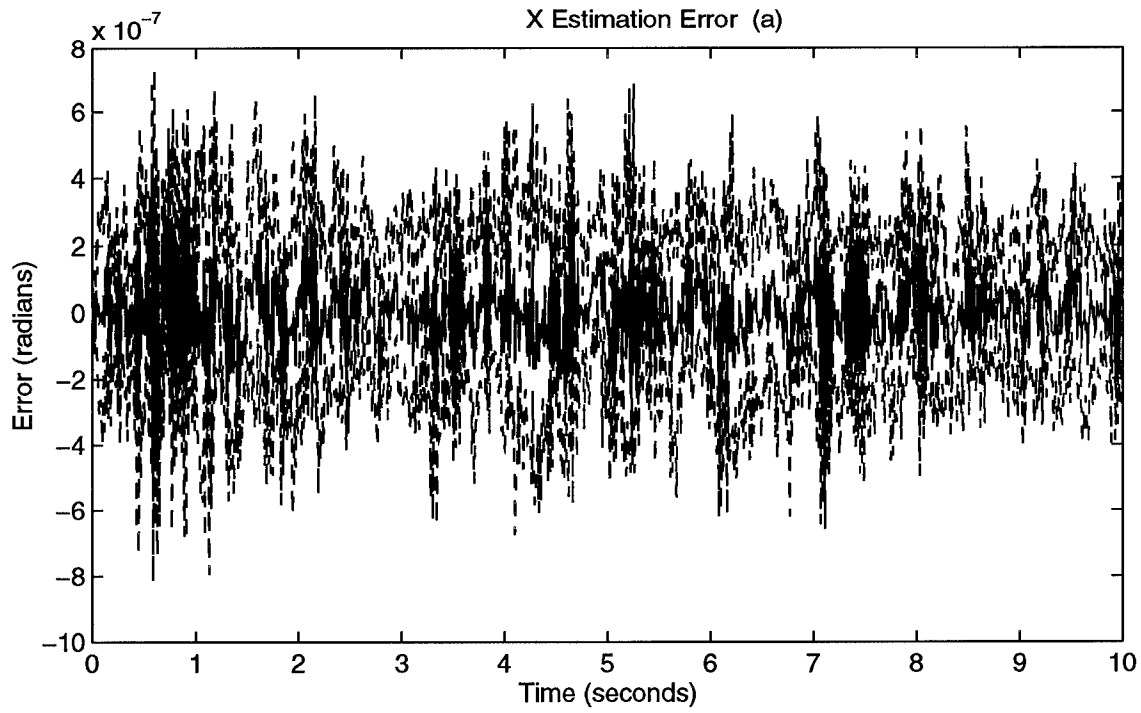


Figure A-53. a_1 and a_3 Moving Up ($a_2 = 1$) Using Residual Monitoring



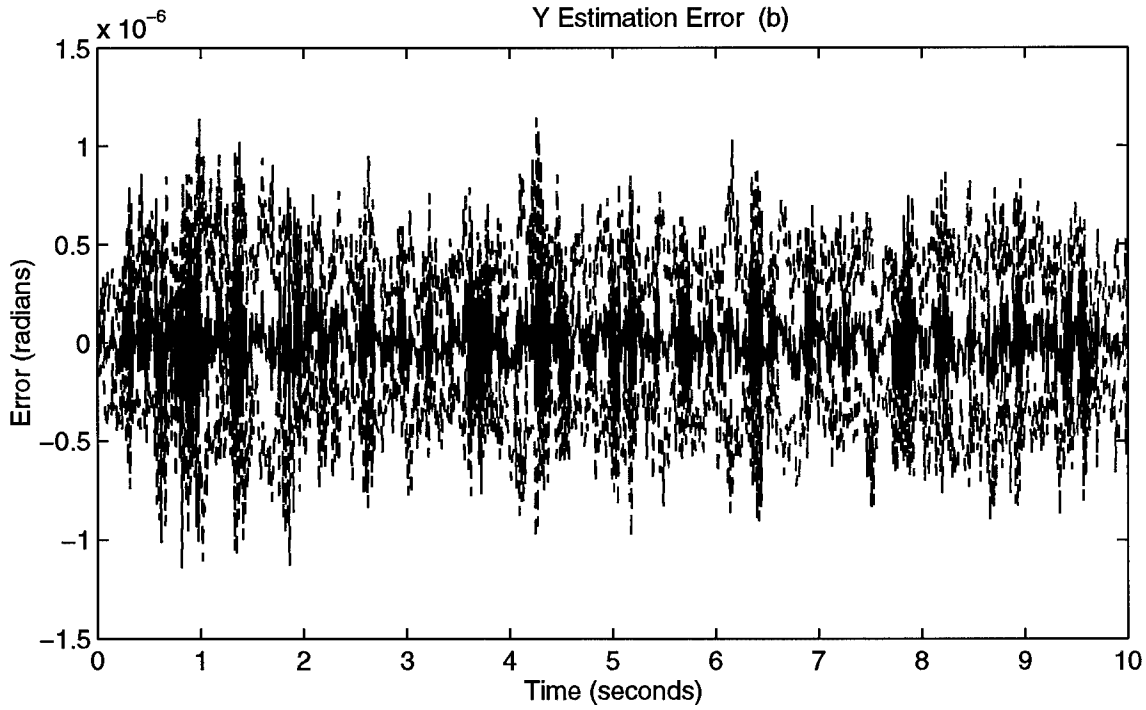
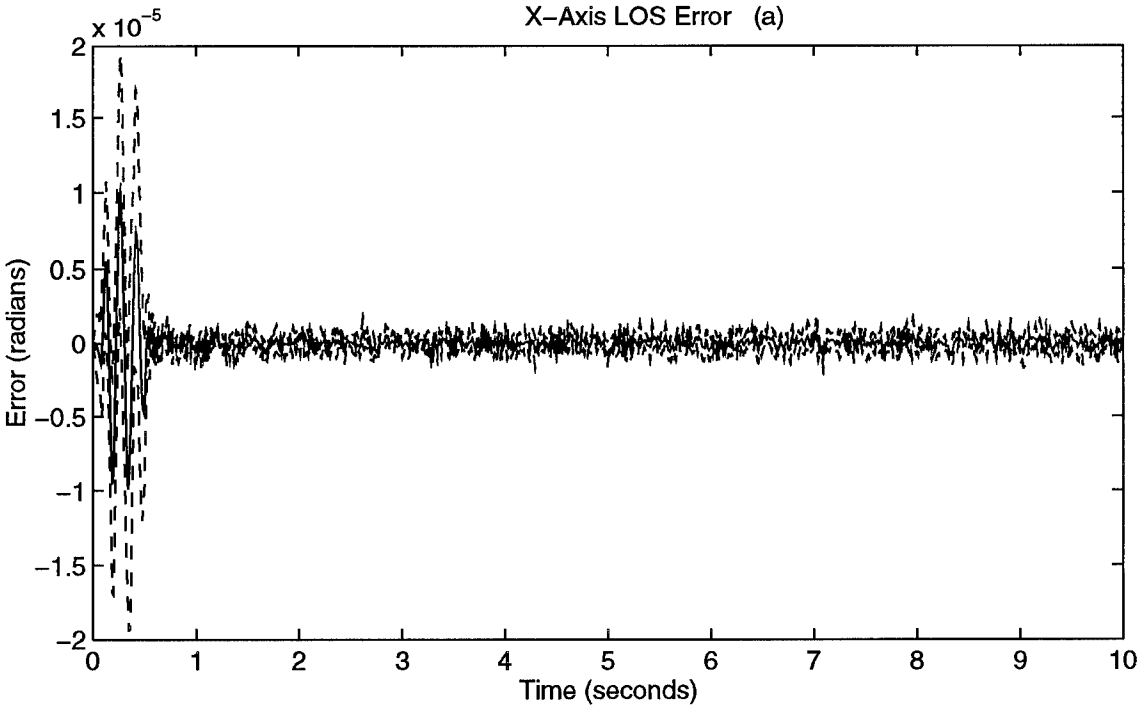


Figure A-54. a_1 and a_3 Moving Up ($a_2 = 1$) Estimation Errors



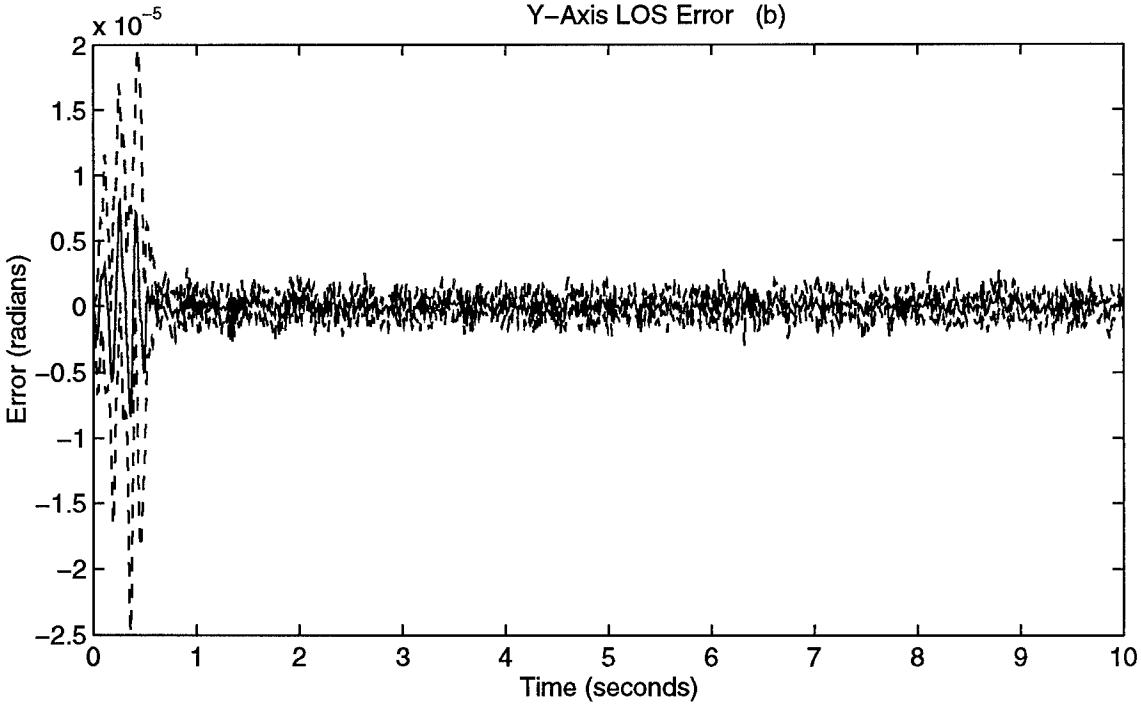


Figure A-55. a_1 and a_3 Moving Up ($a_2 = 1$) LOS Errors

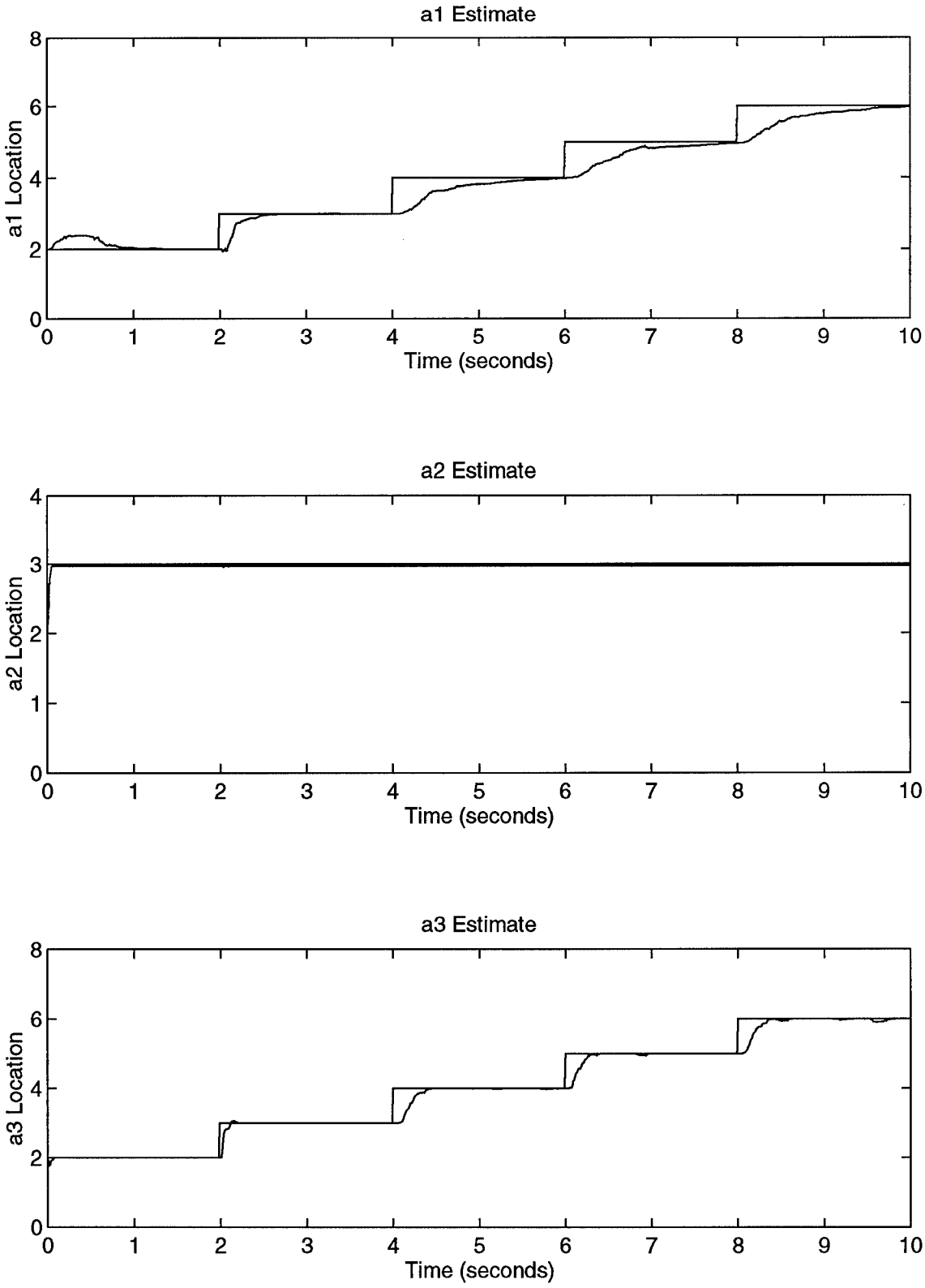


Figure A-56. a_1 and a_3 Moving Up ($a_2 = 3$) Using Parameter Position Monitoring

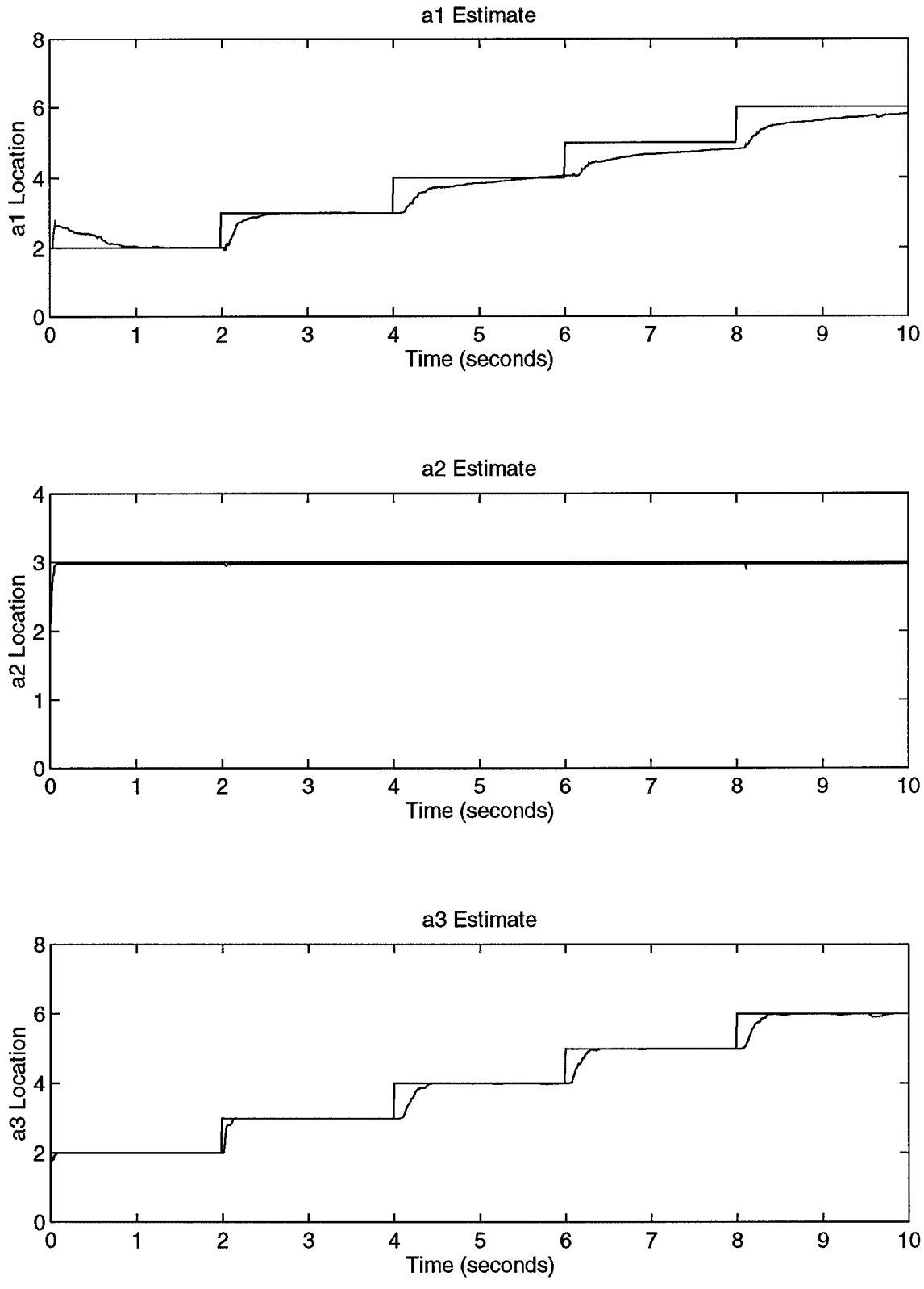


Figure A-57. a_1 and a_3 Moving Up ($a_2 = 3$) Using Probability Monitoring

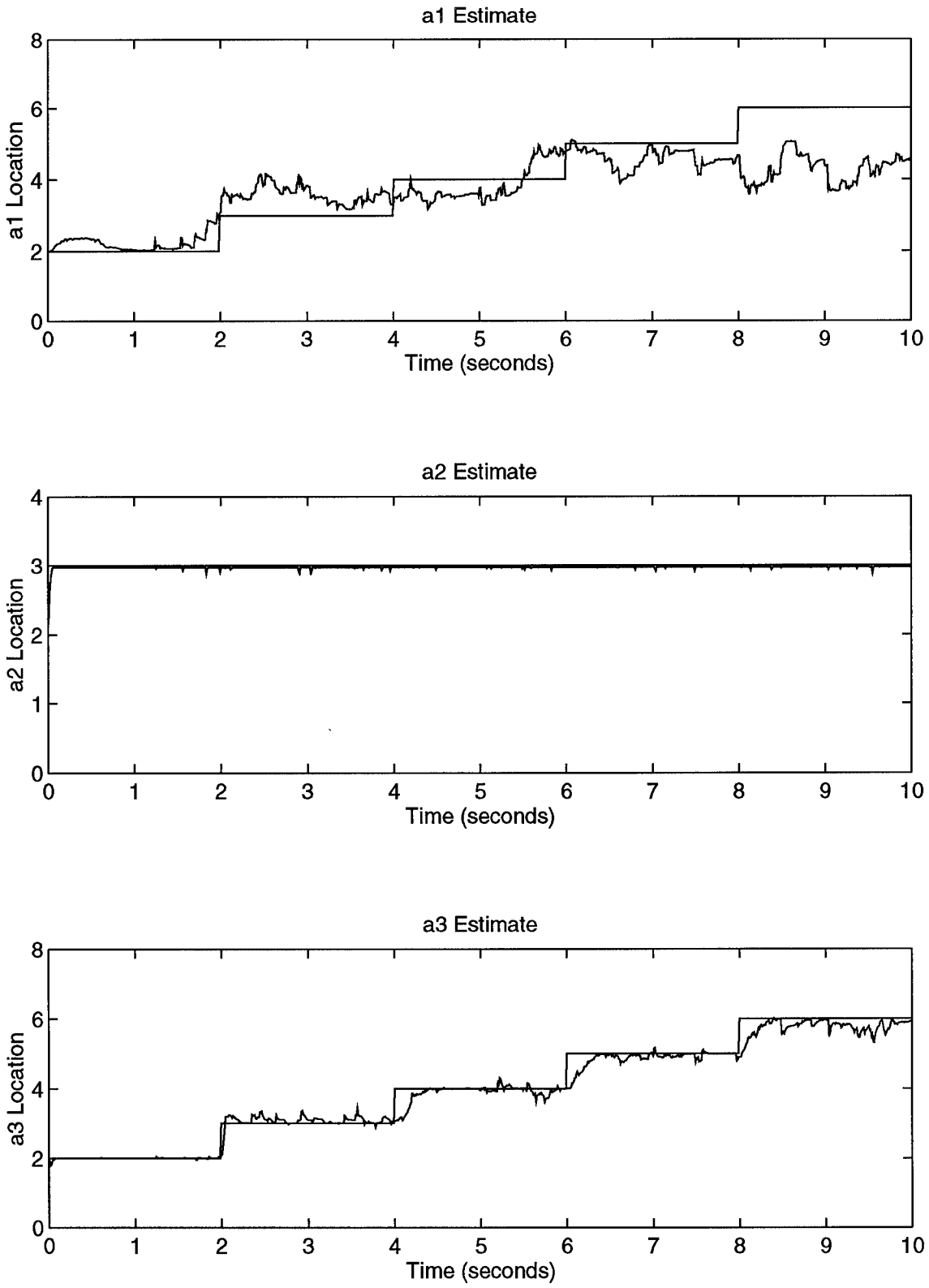
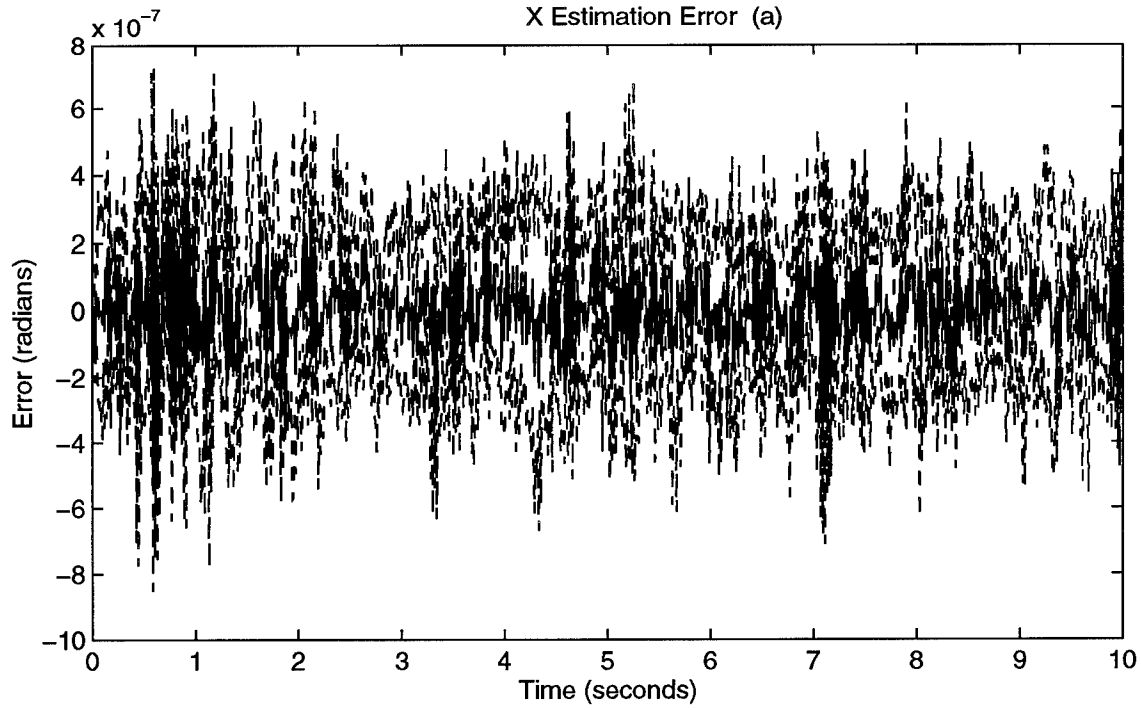


Figure A-58. a_1 and a_3 Moving Up ($a_2 = 3$) Using Residual Monitoring



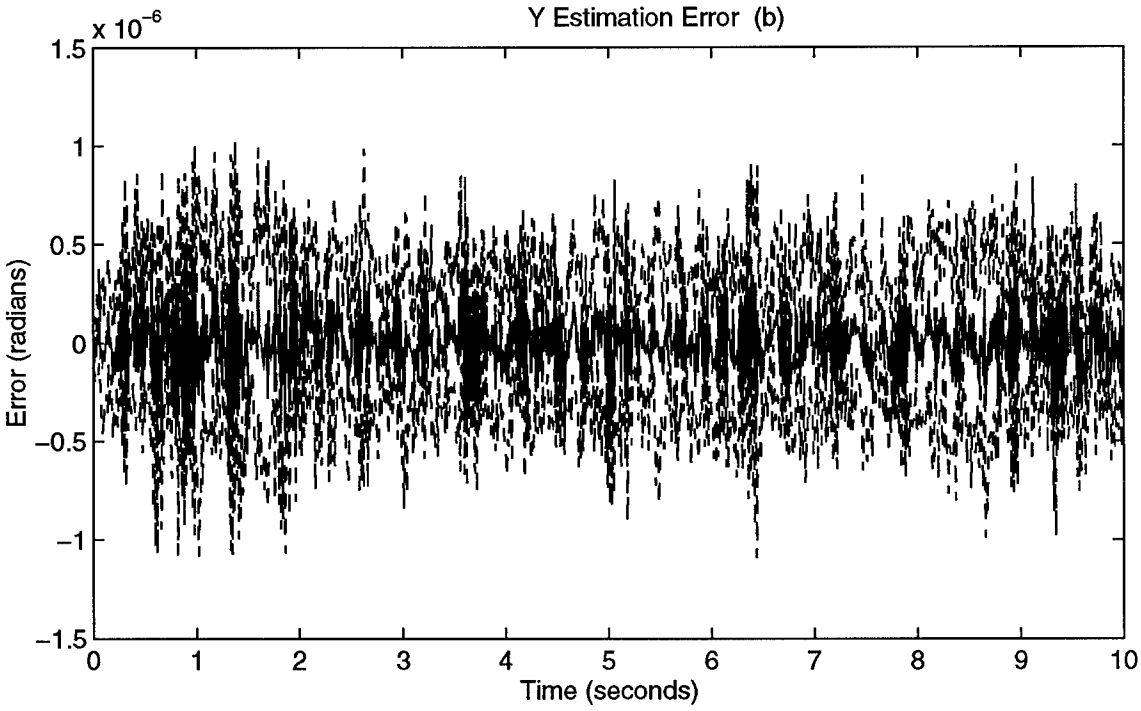
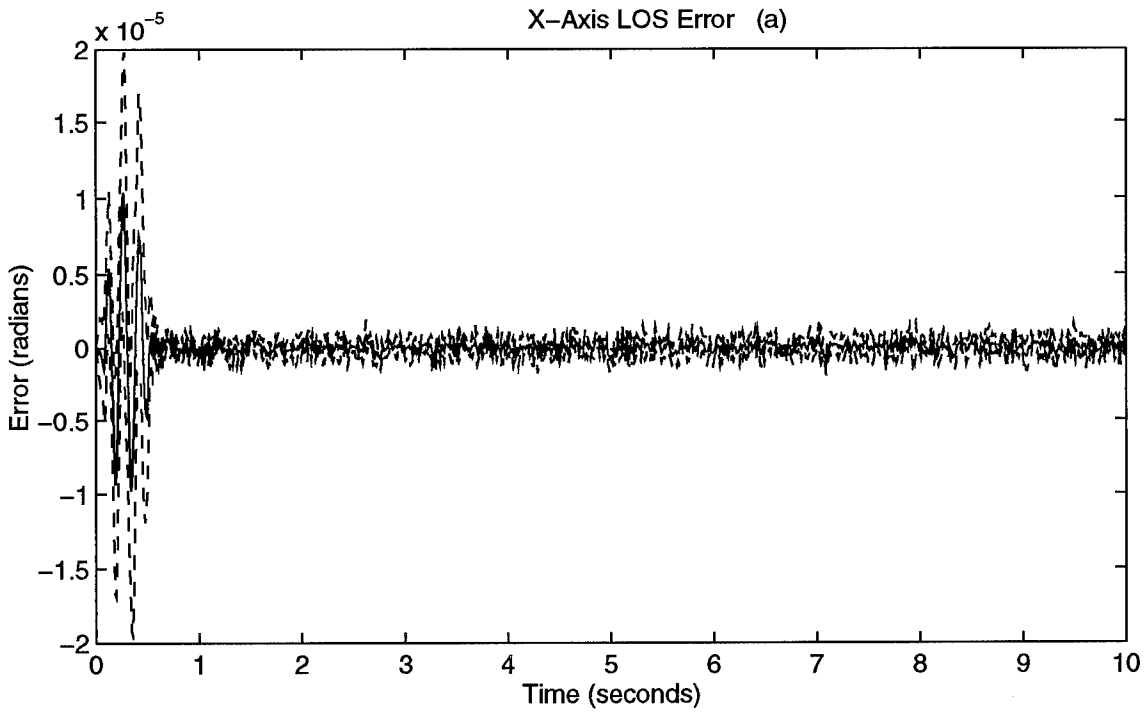


Figure A-59. a_1 and a_3 Moving Up ($a_2 = 3$) Estimation Errors



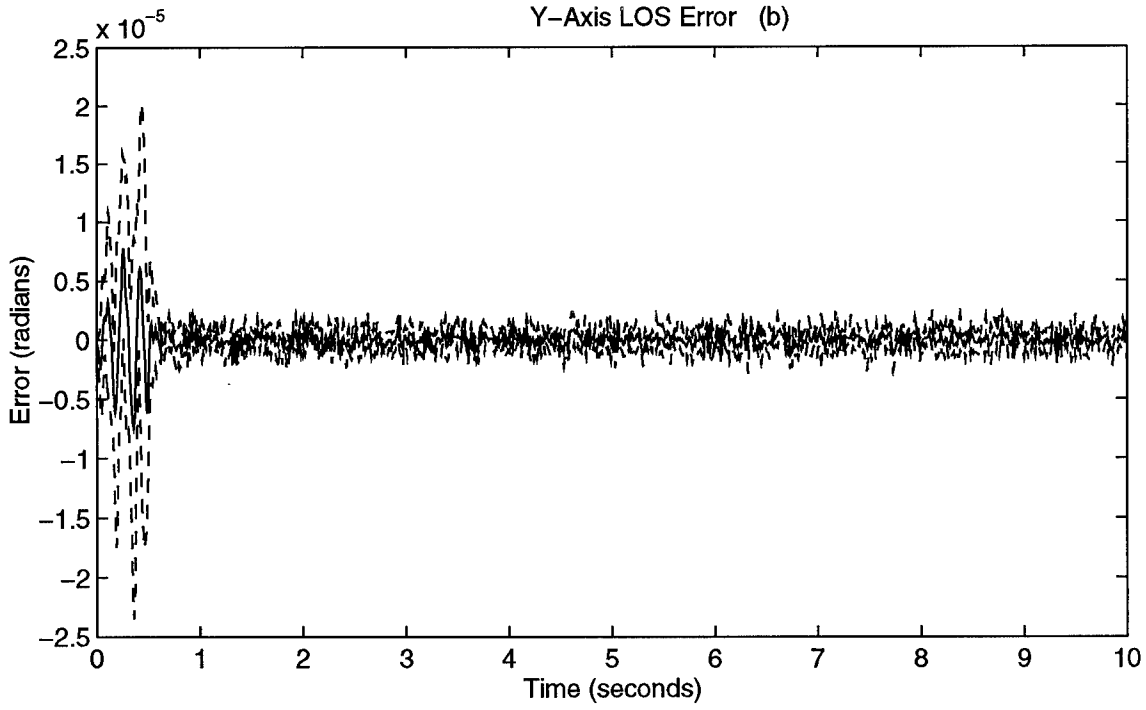


Figure A-60. a_1 and a_3 Moving Up ($a_2 = 3$) LOS Errors

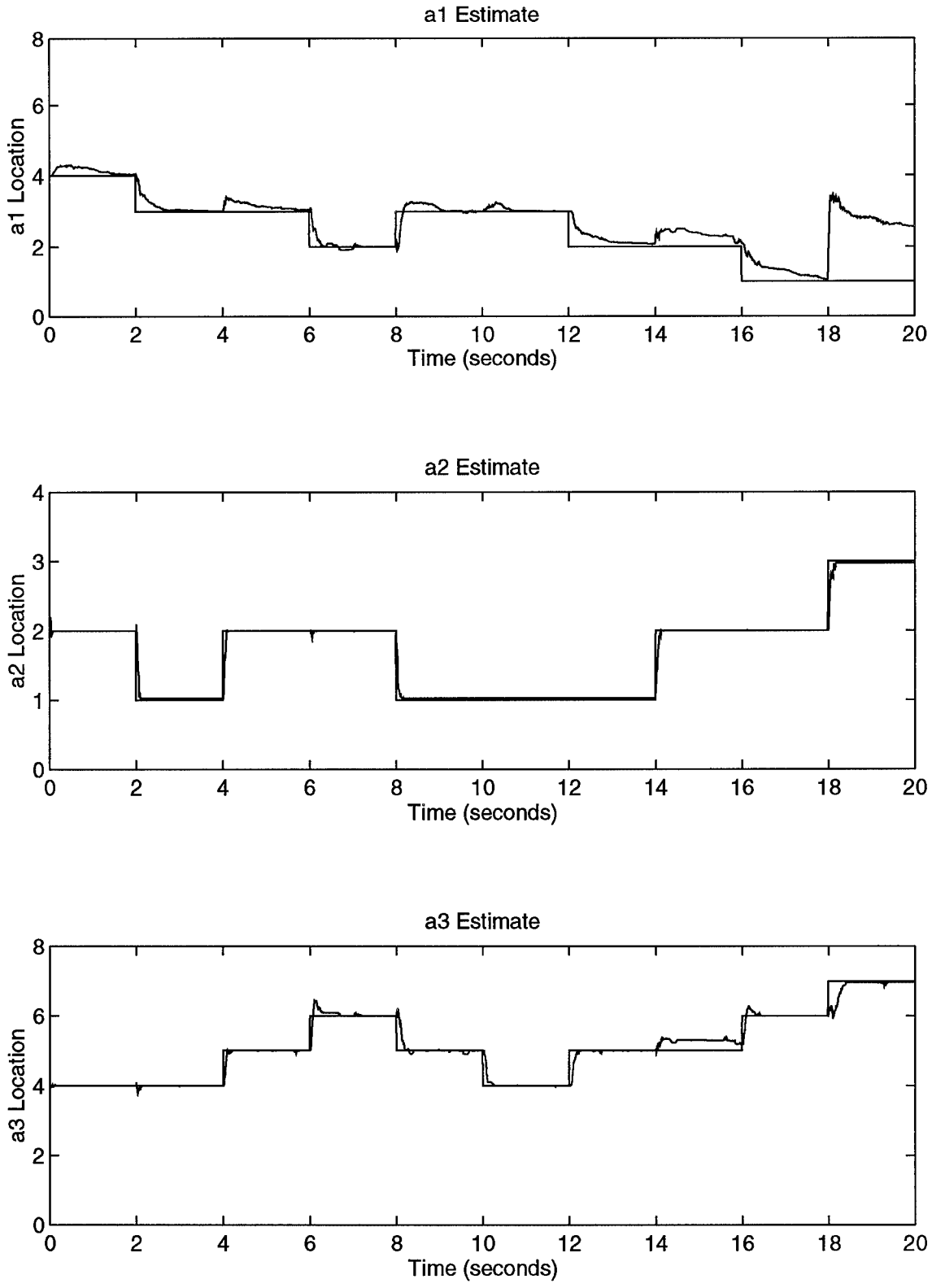


Figure A-61. Random Movement #1 Using Parameter Position Monitoring

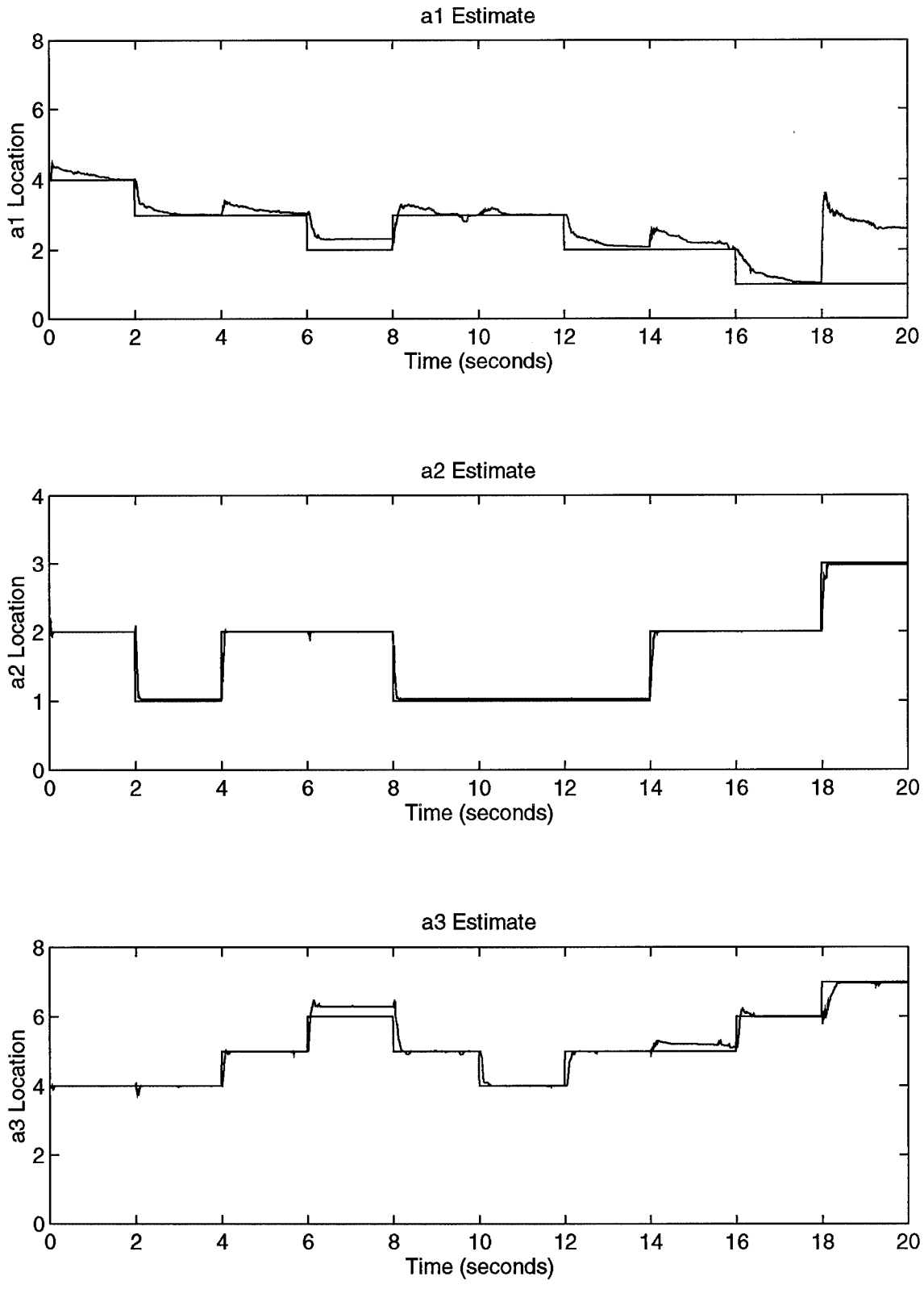


Figure A-62. Random Movement #1 Using Probability Monitoring

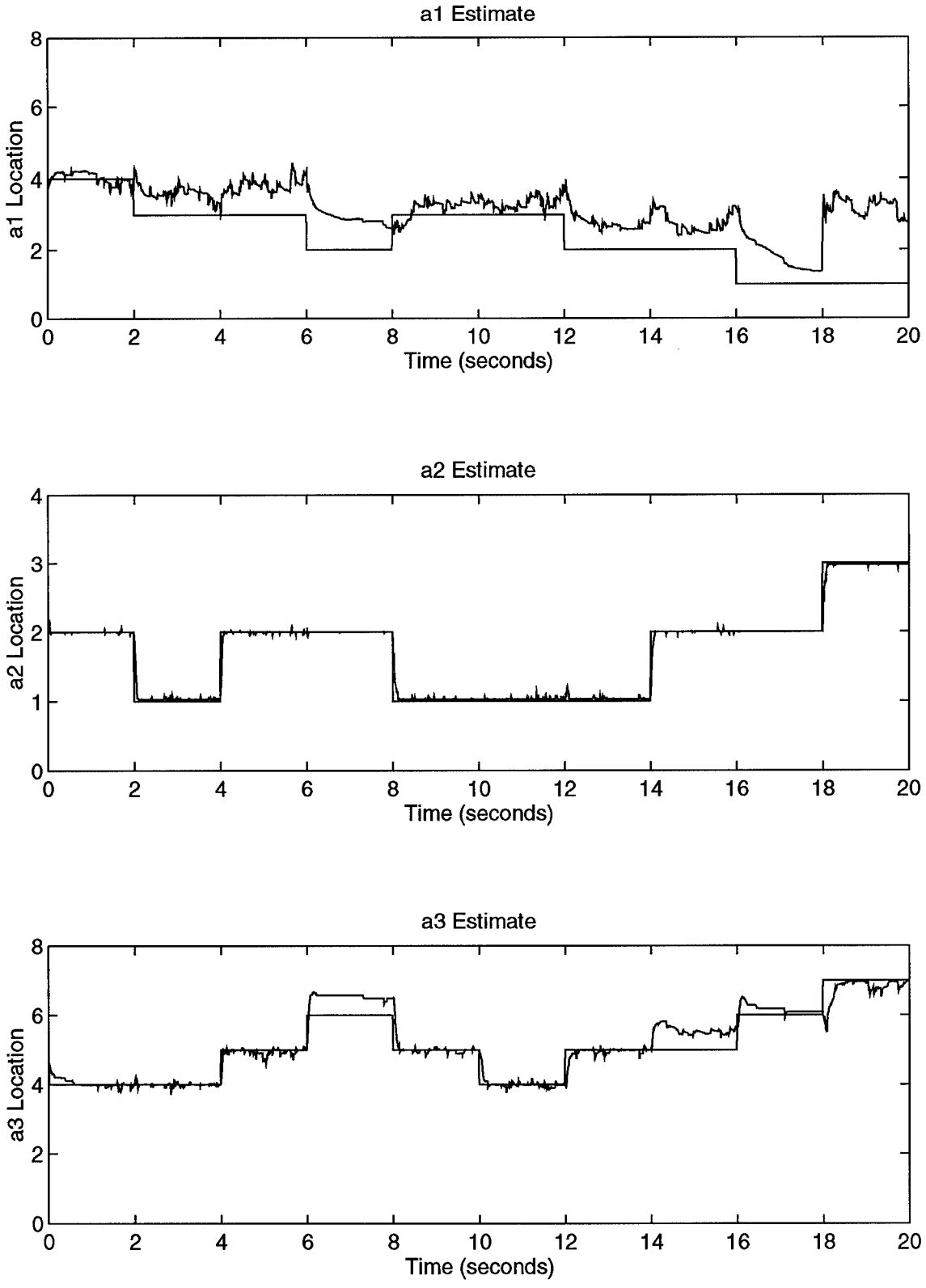


Figure A-63. Random Movement #1 Using Residual Monitoring

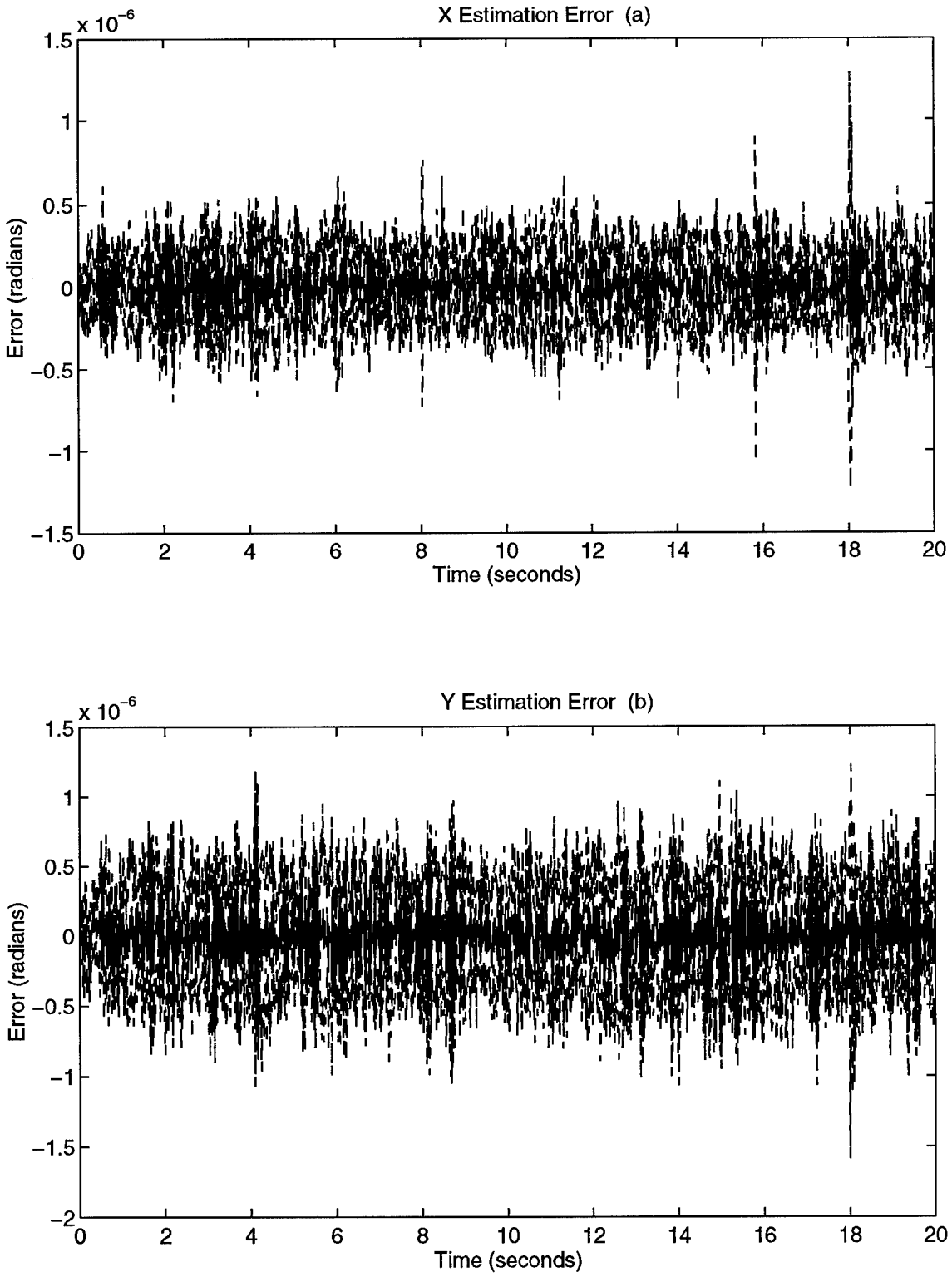
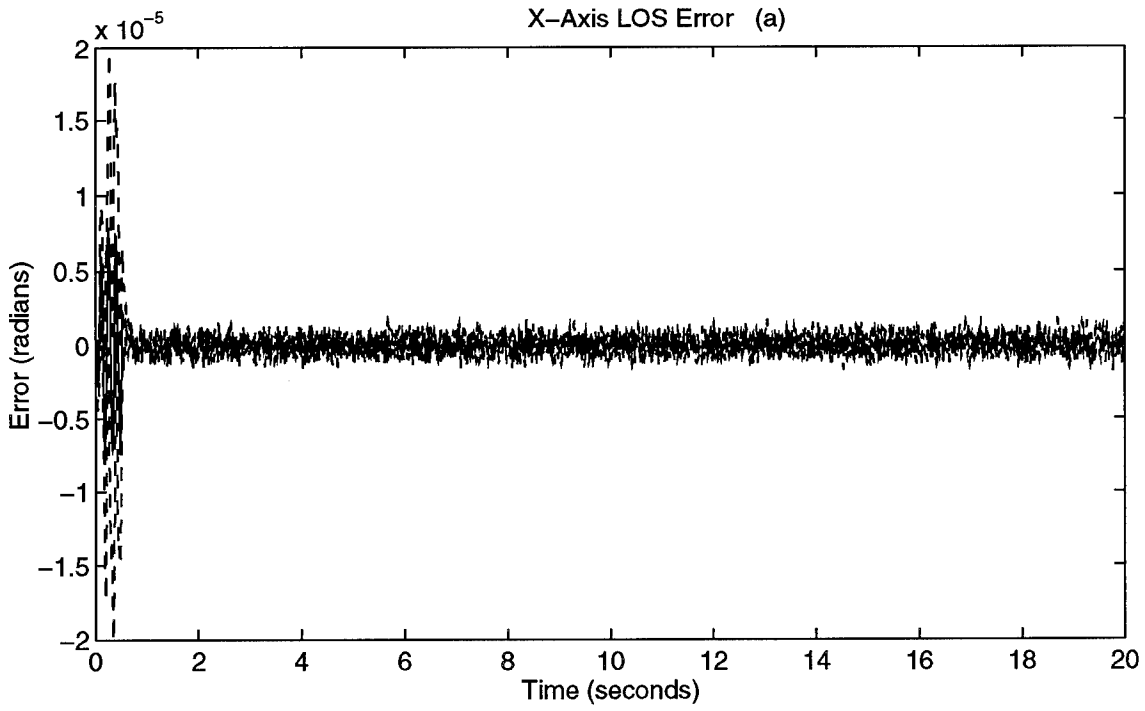


Figure A-64. Random Movement #1 Estimation Errors



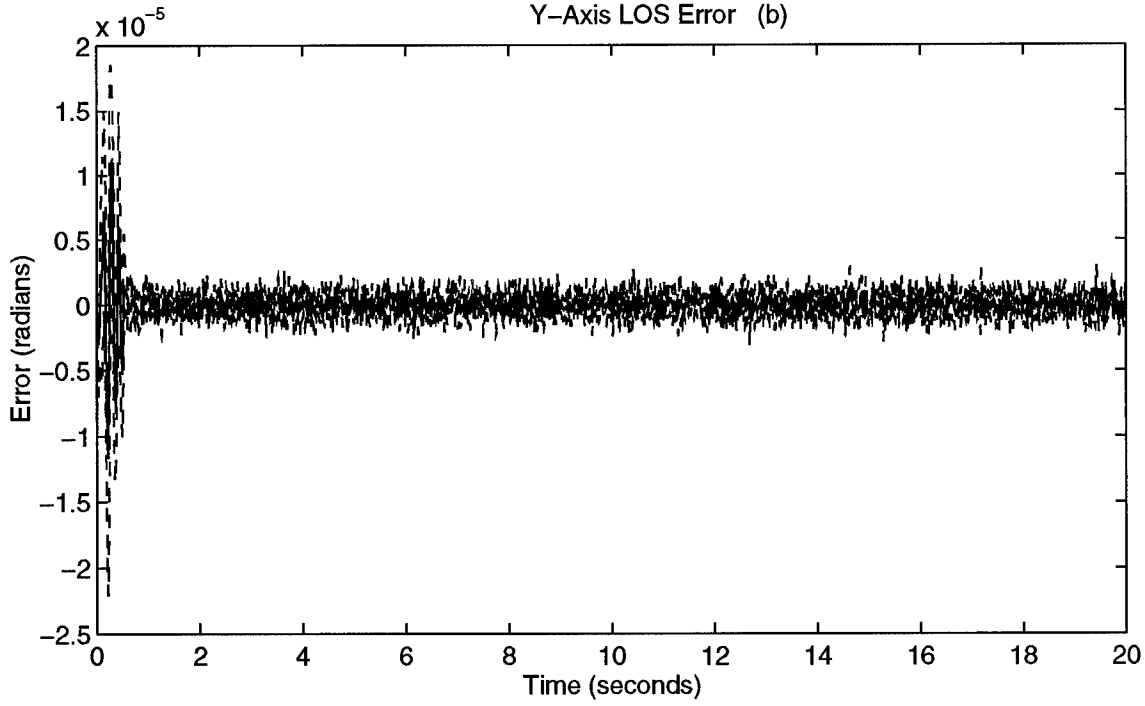


Figure A-65. Random Movement #1 LOS Errors

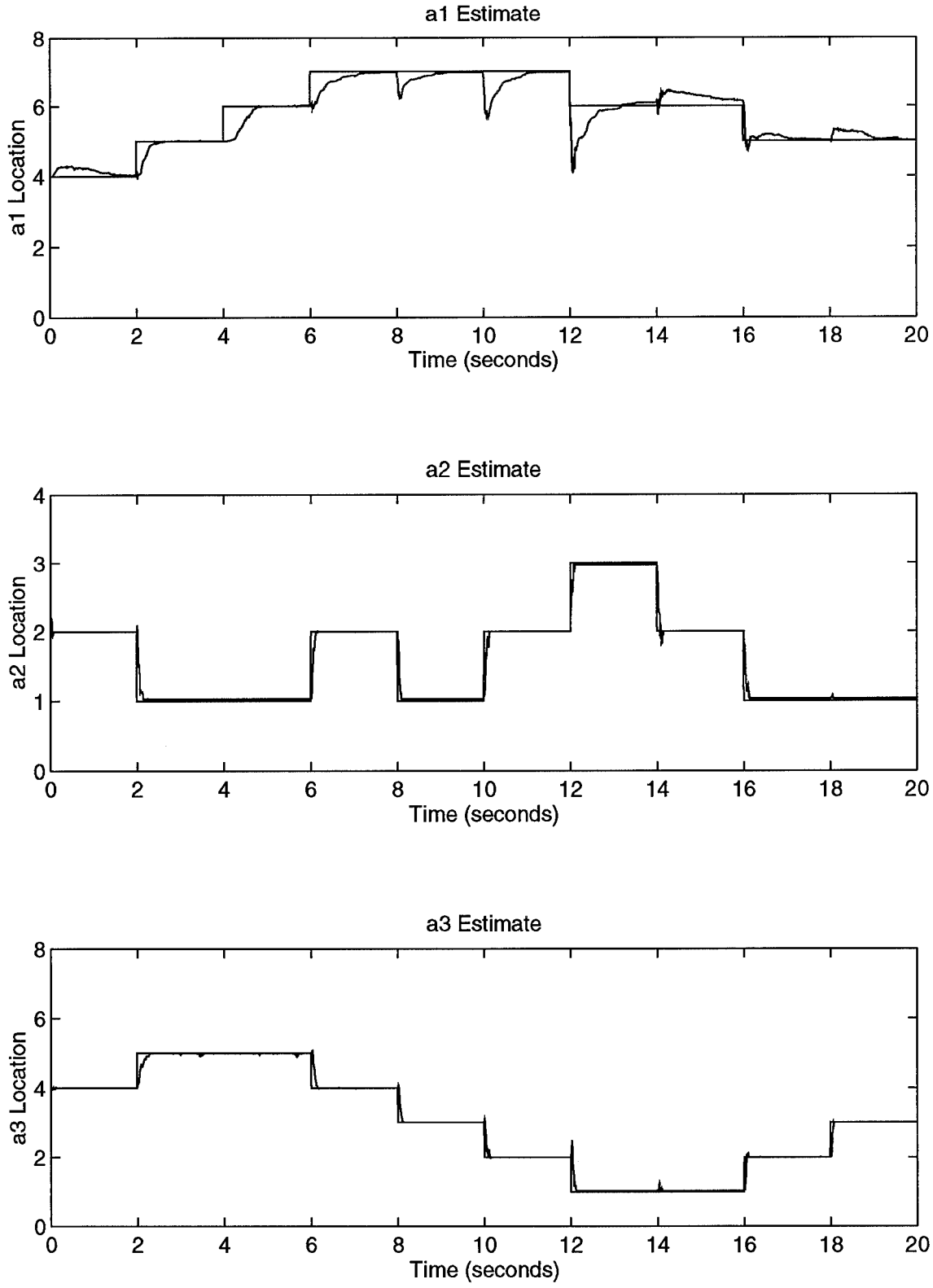


Figure A-66. Random Movement #2 Using Parameter Position Monitoring

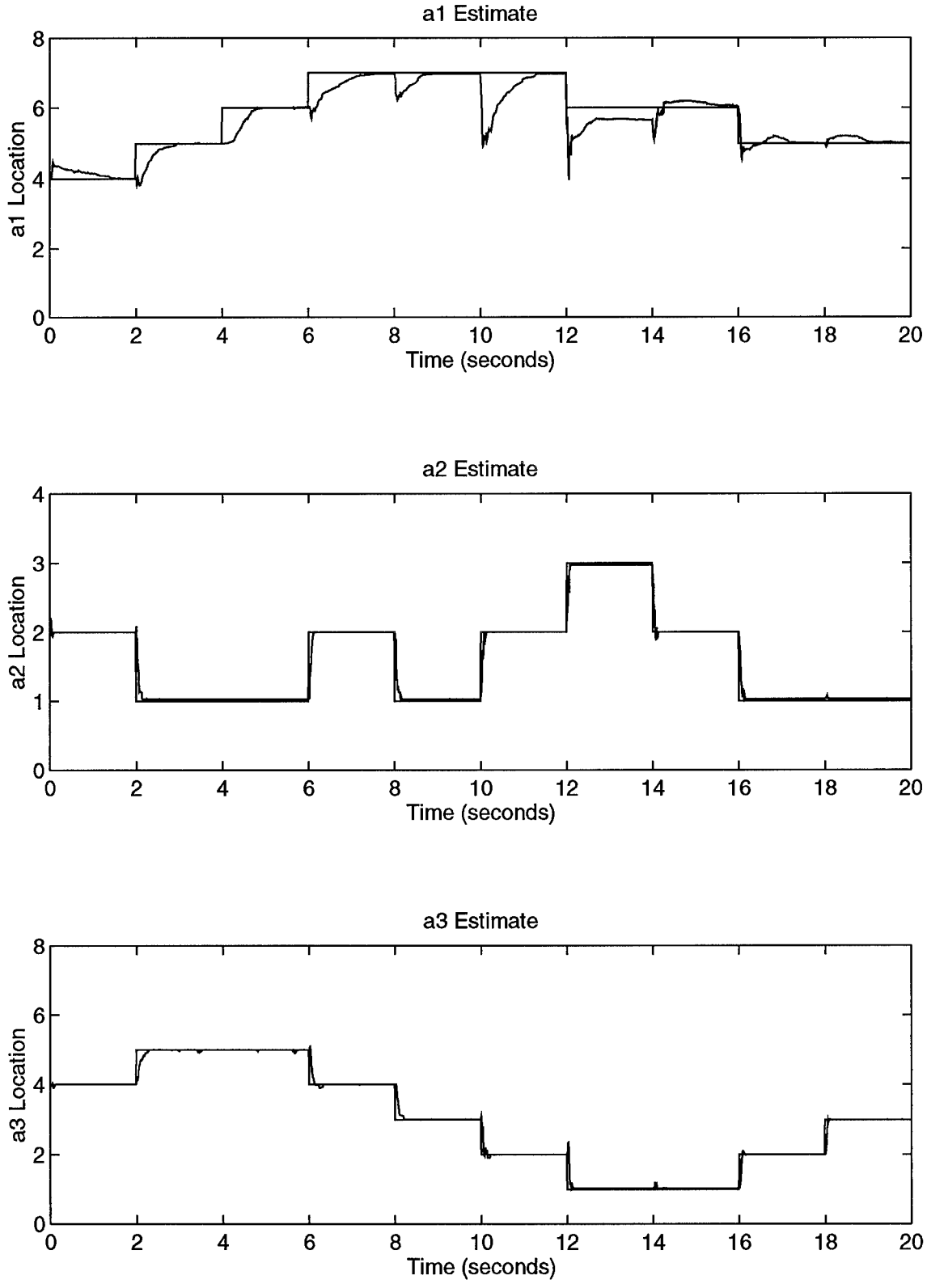


Figure A-67. Random Movement #2 Using Probability Monitoring

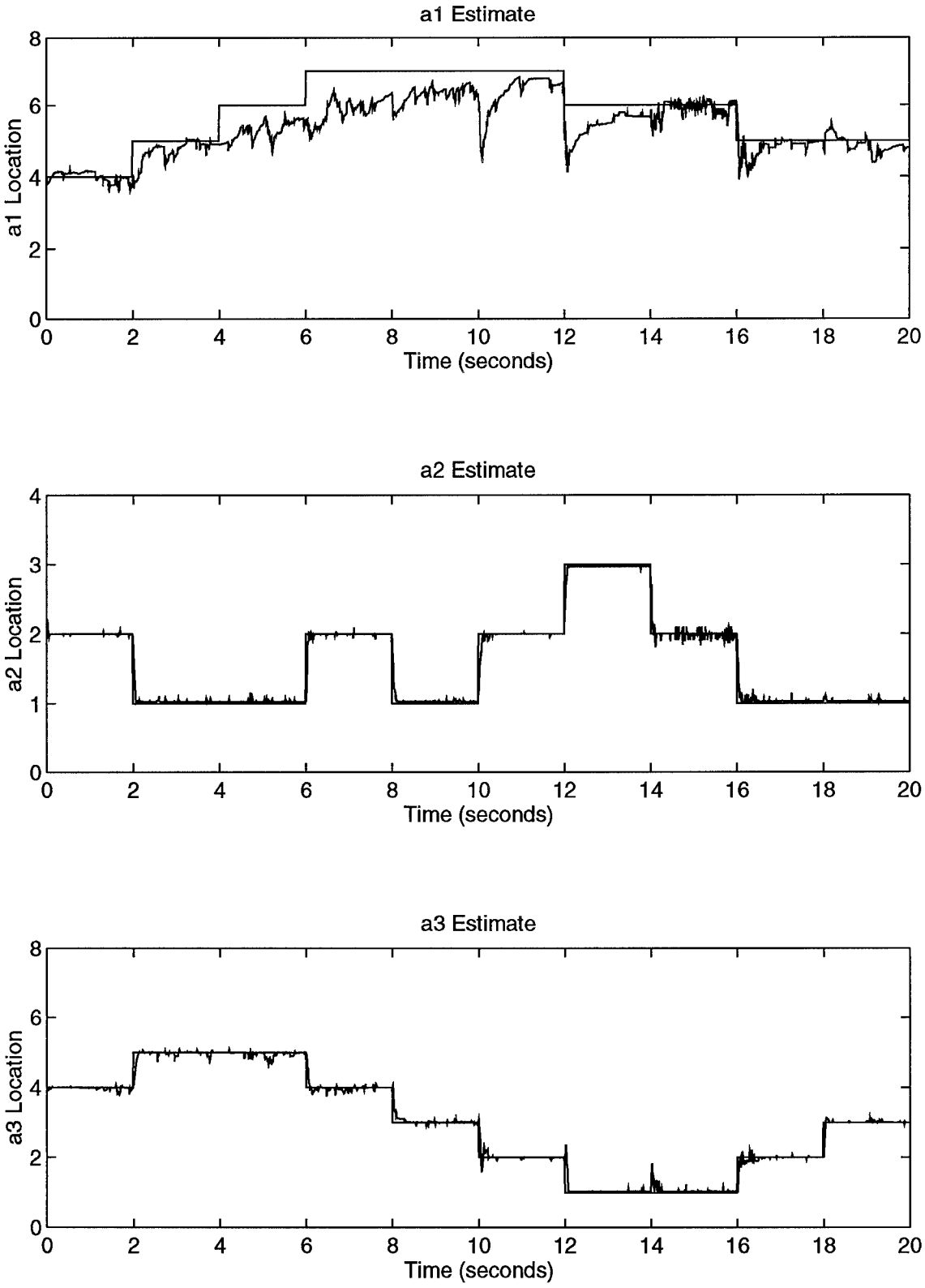


Figure A-68. Random Movement #2 Using Residual Monitoring

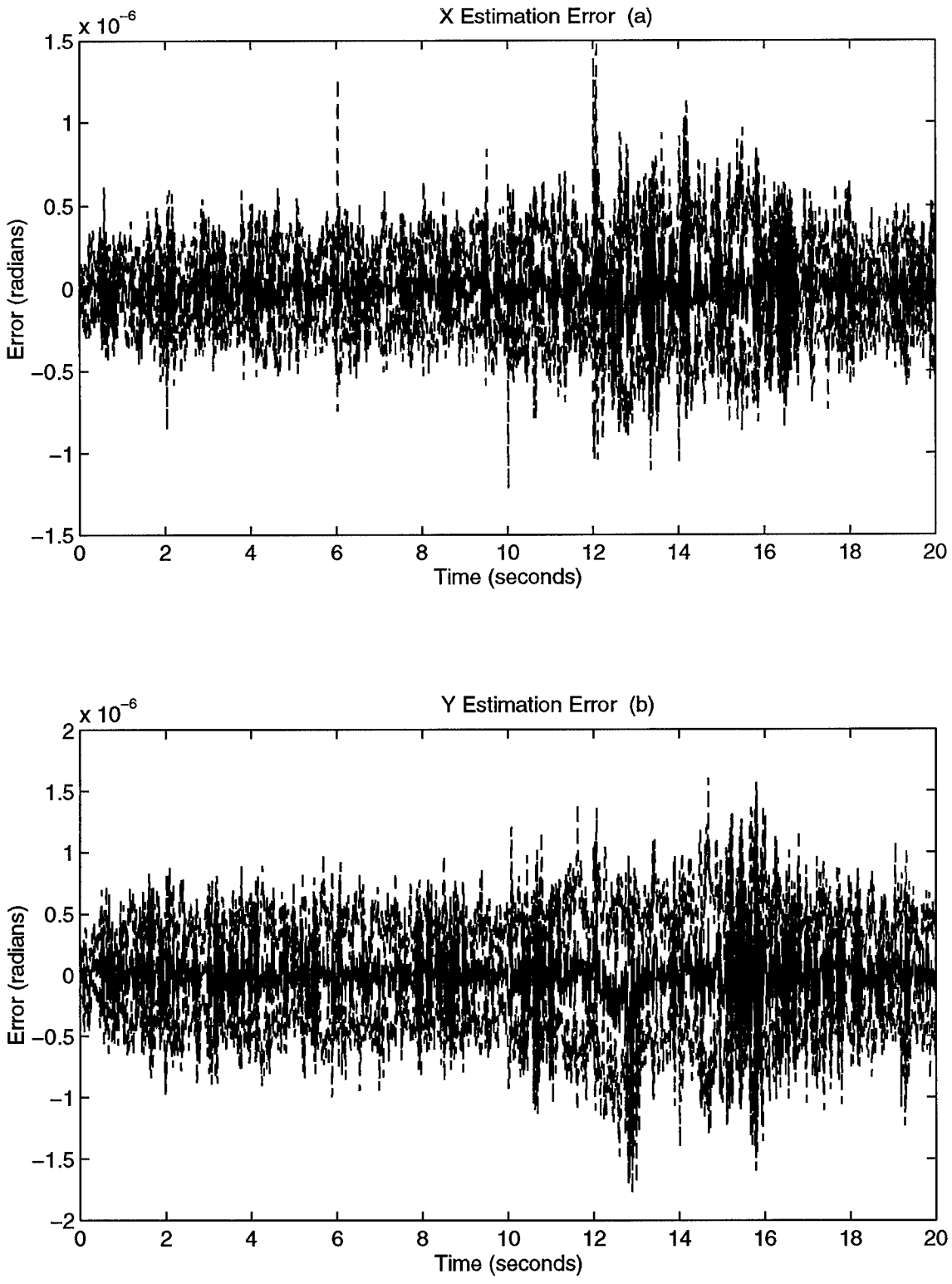
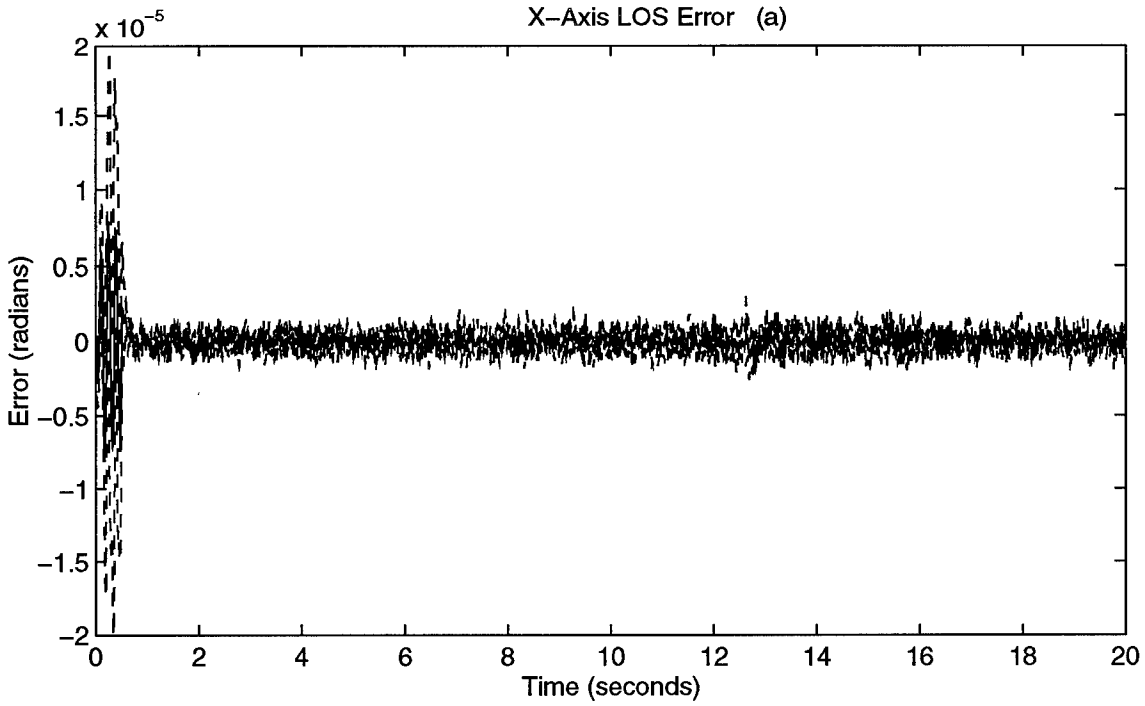


Figure A-69. Random Movement #2 Estimation Errors



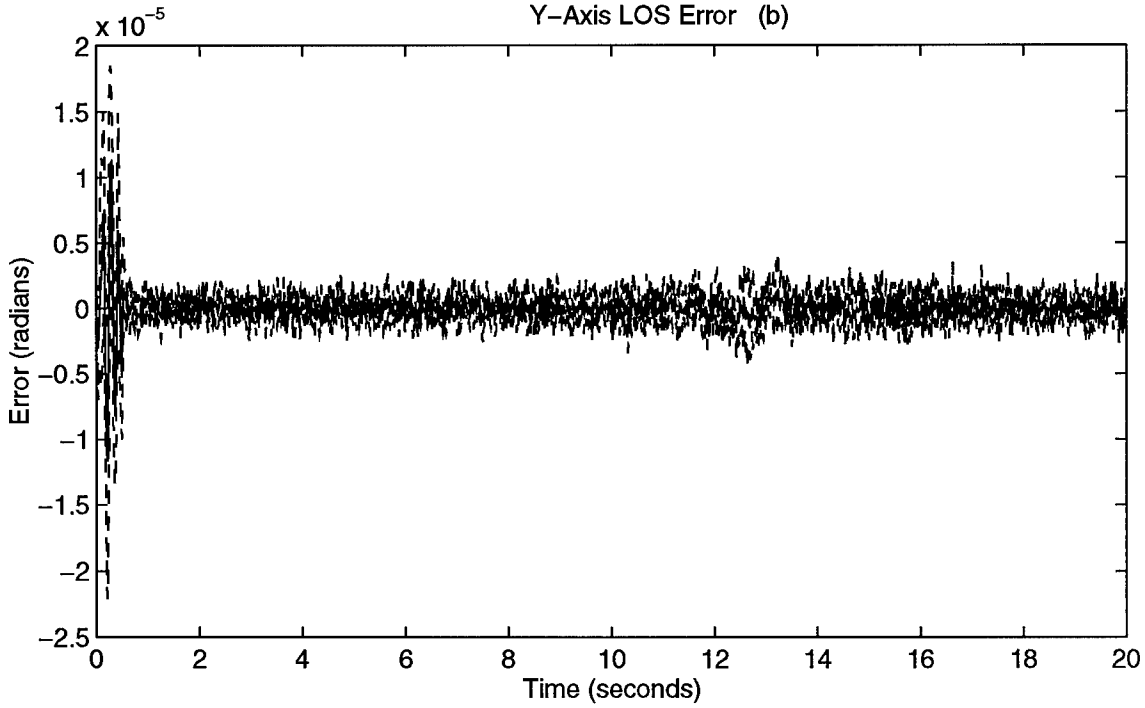


Figure A-70. Random Movement #2 LOS Errors

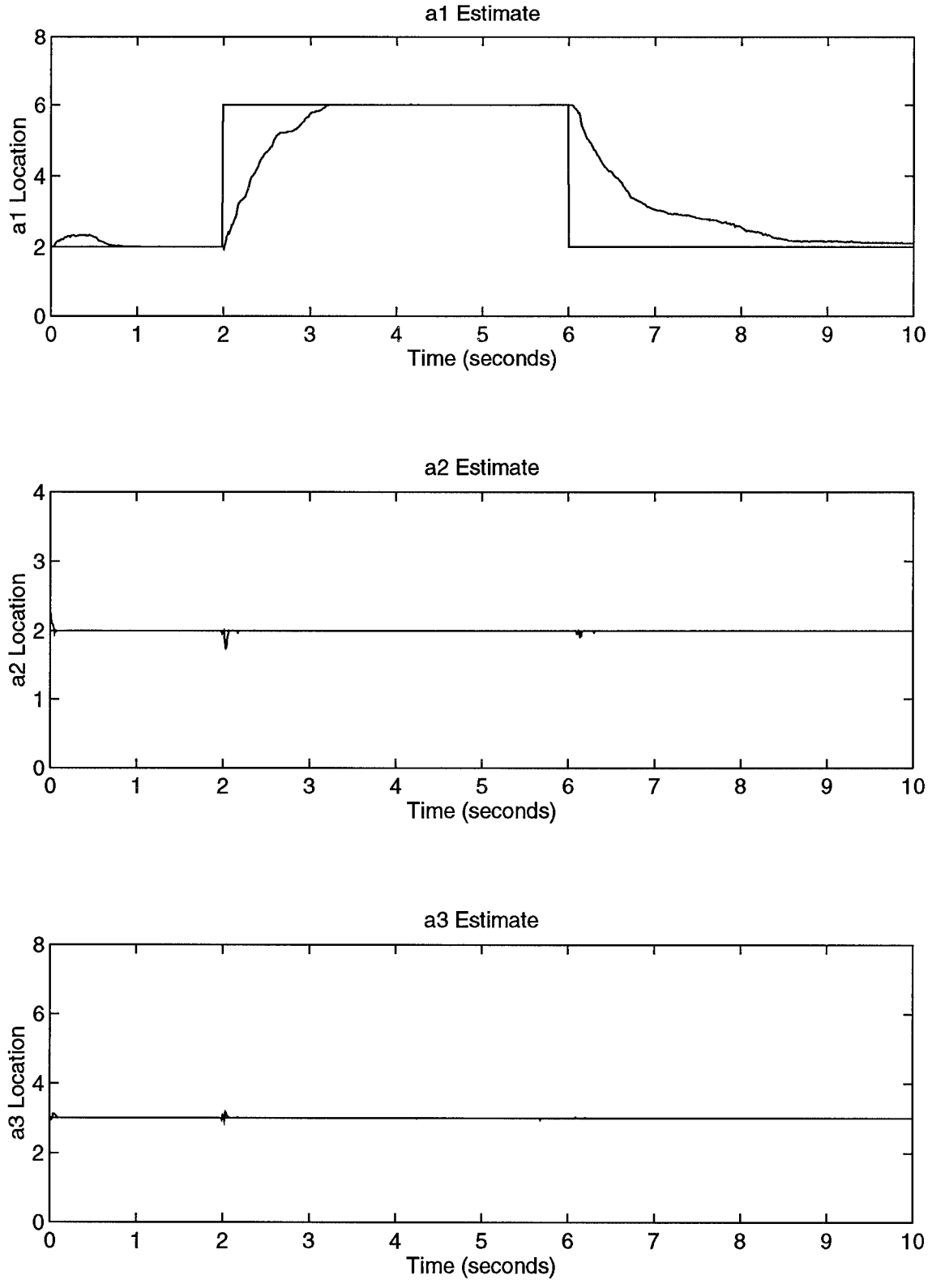


Figure A-71. a_1 Parameter Jumps Using Parameter Position Monitoring

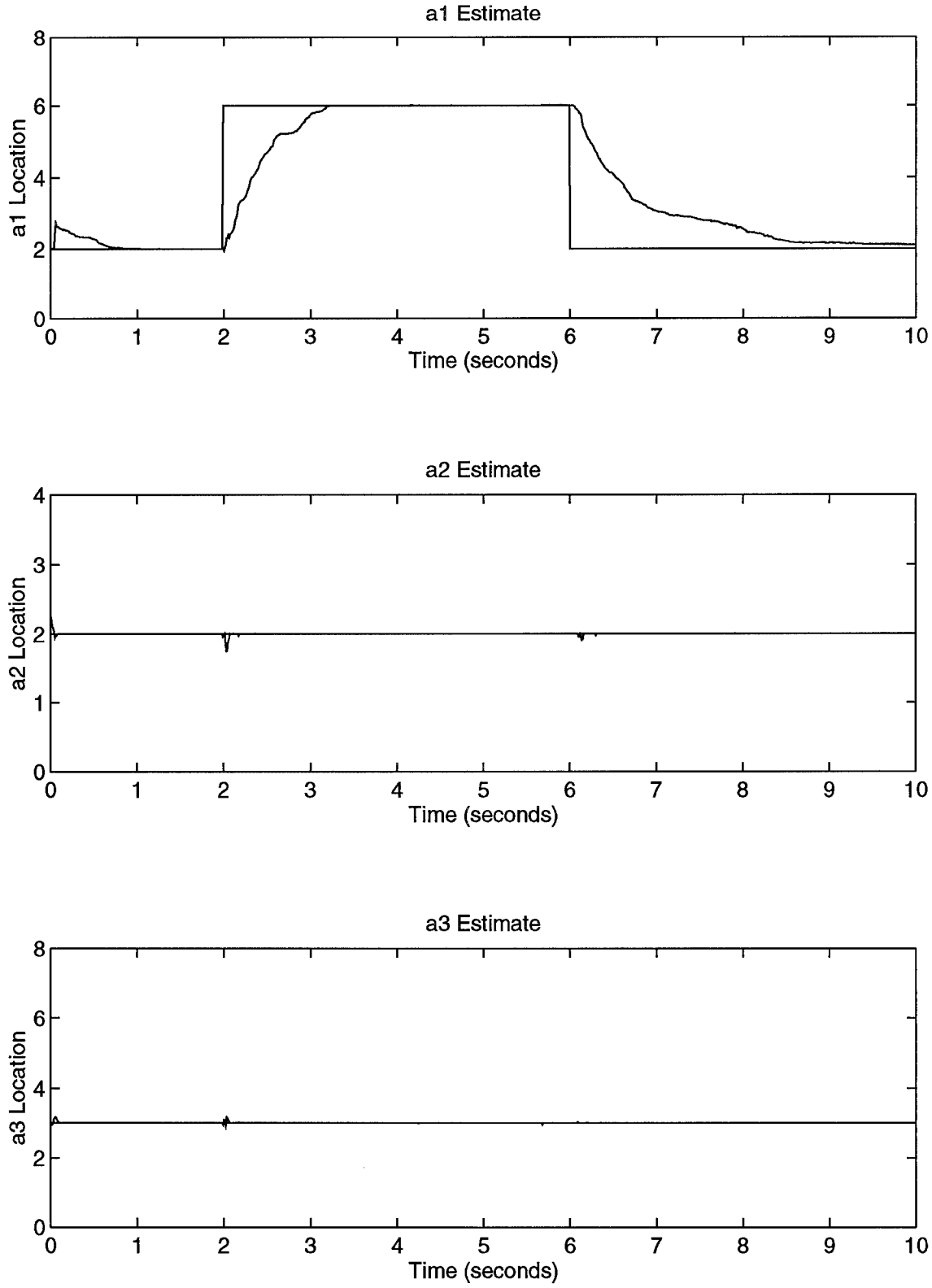


Figure A-72. a_1 Parameter Jumps Using Probability Monitoring

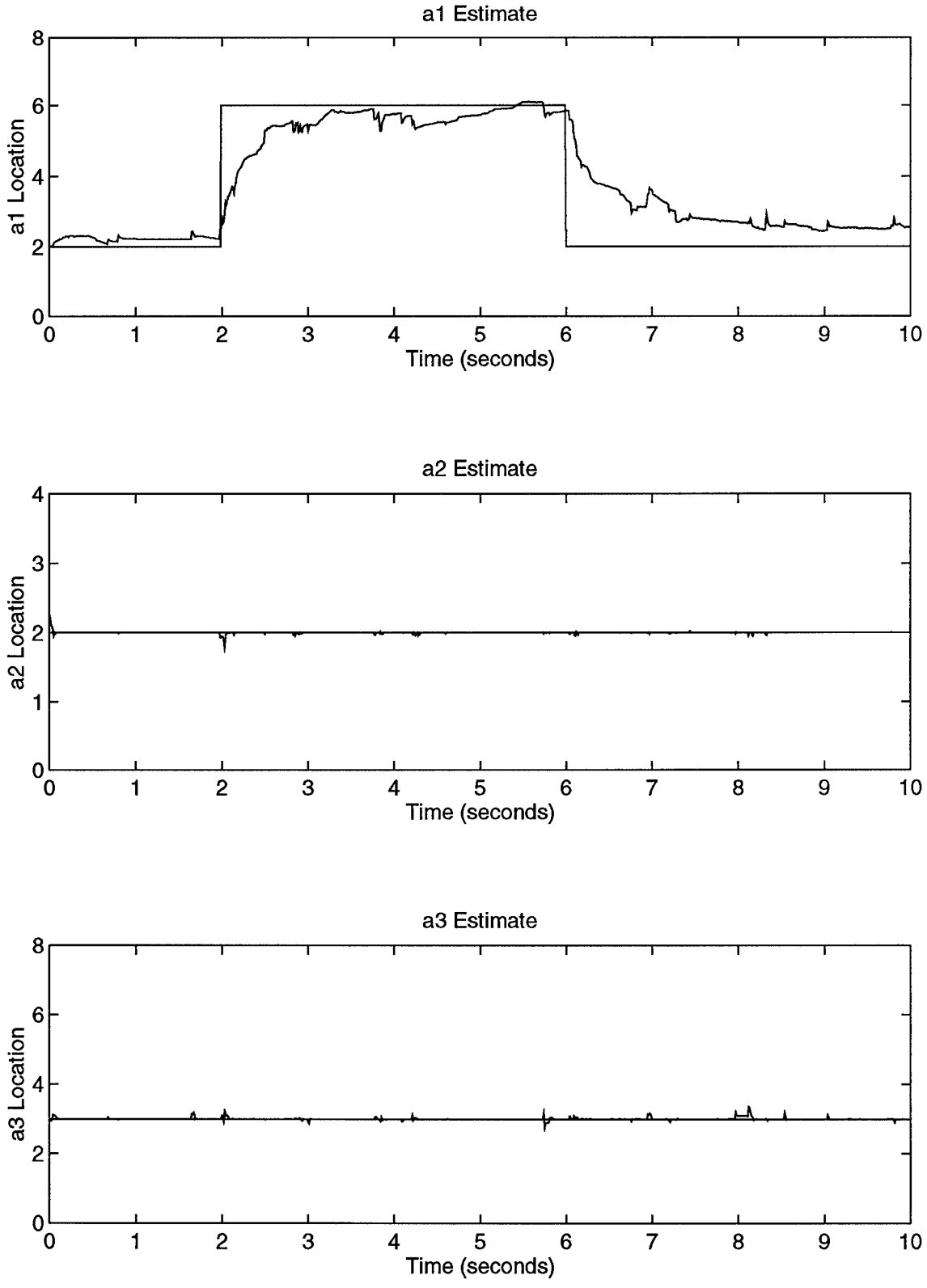
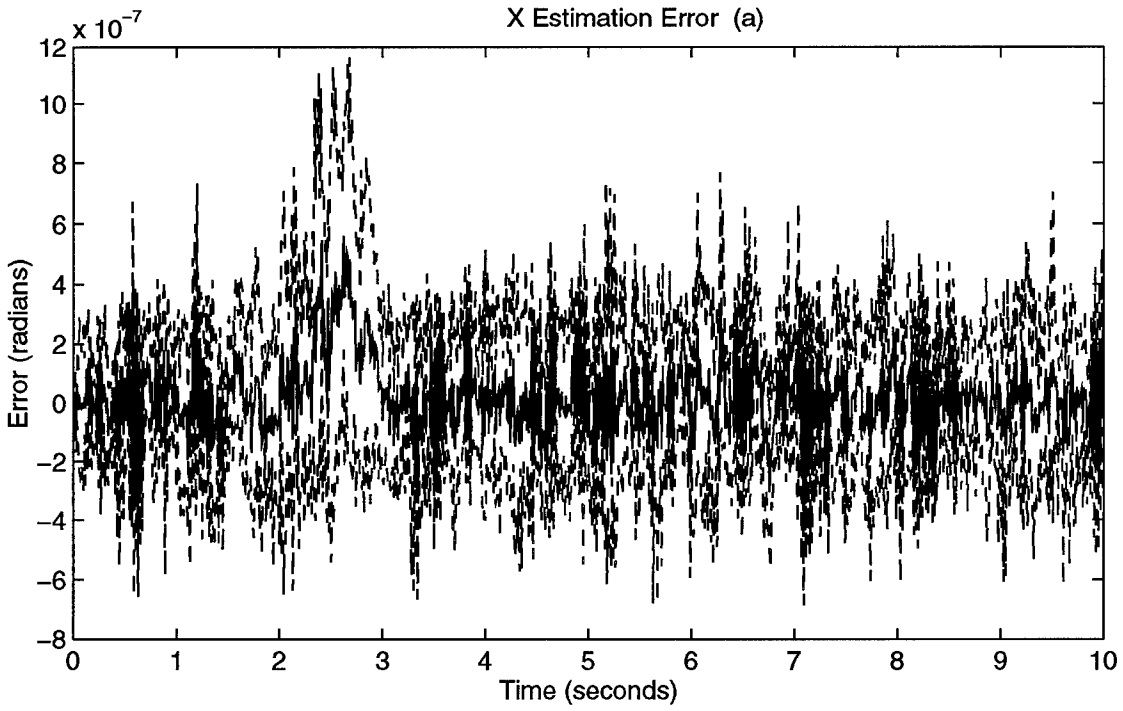


Figure A-73. a_1 Parameter Jumps Using Residual Monitoring



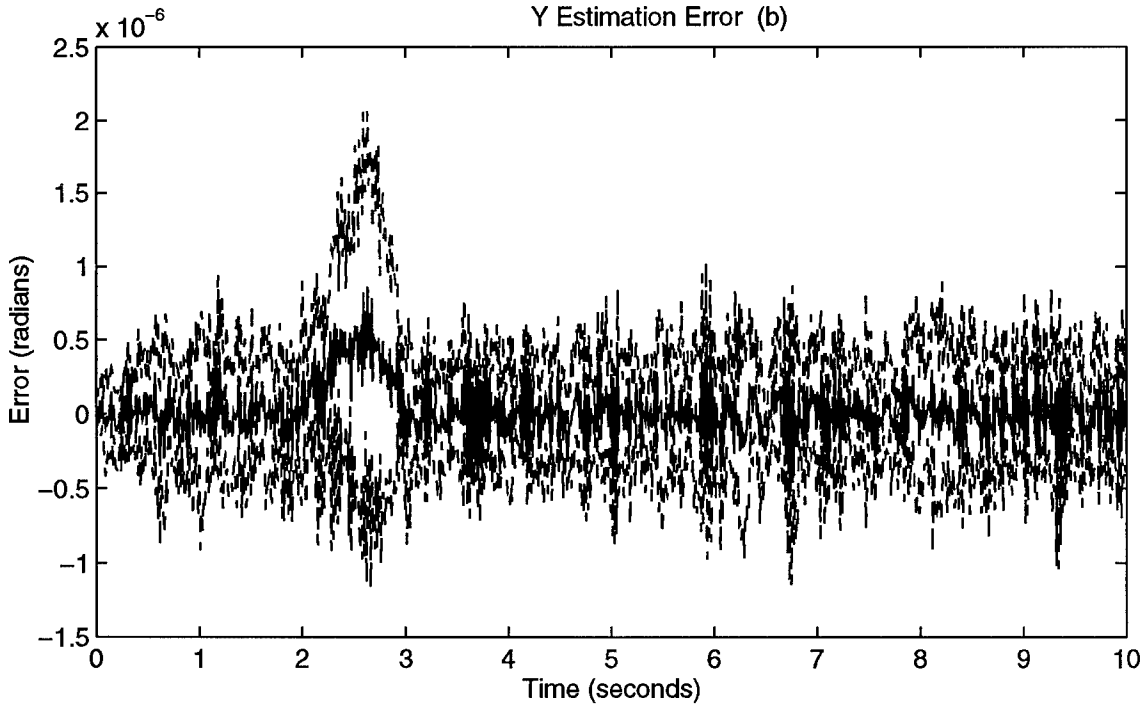
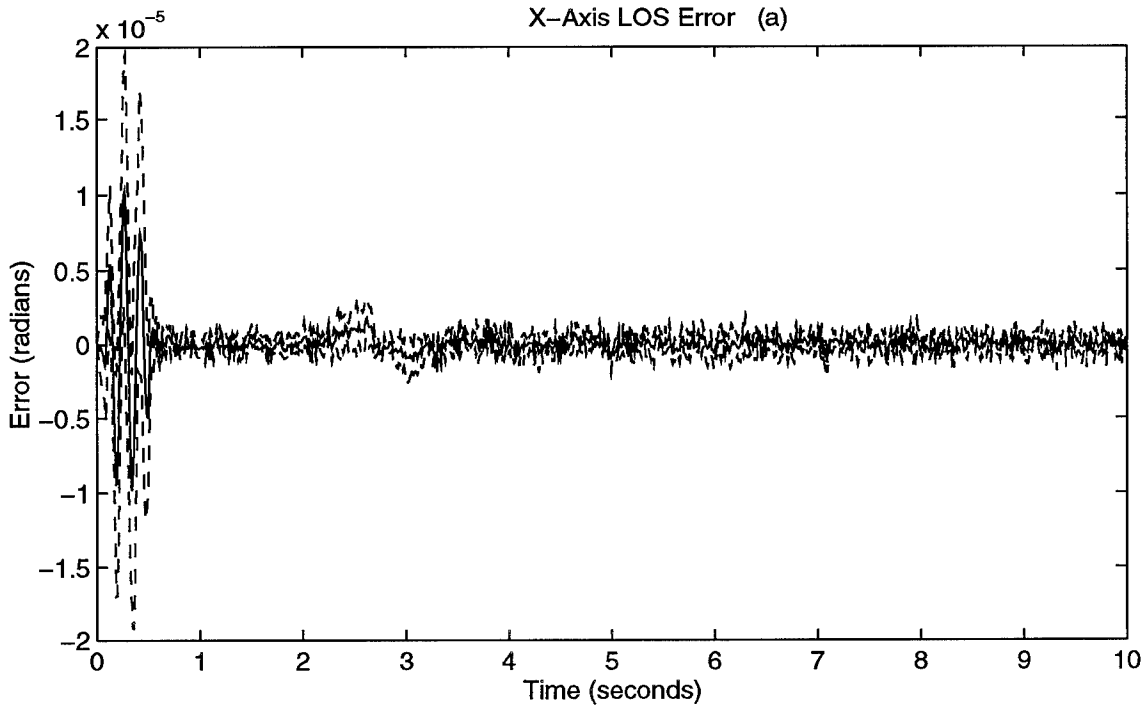


Figure A-74. a_1 Parameter Jumps Estimation Errors



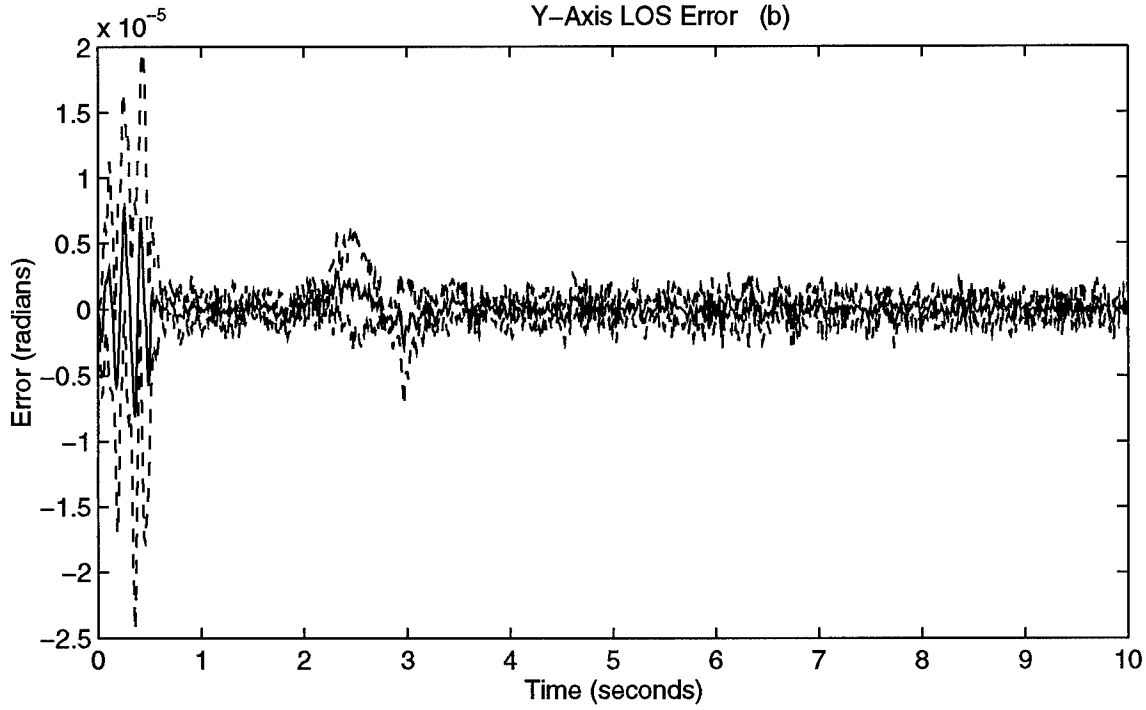


Figure A-75. a_1 Parameter Jumps LOS Errors

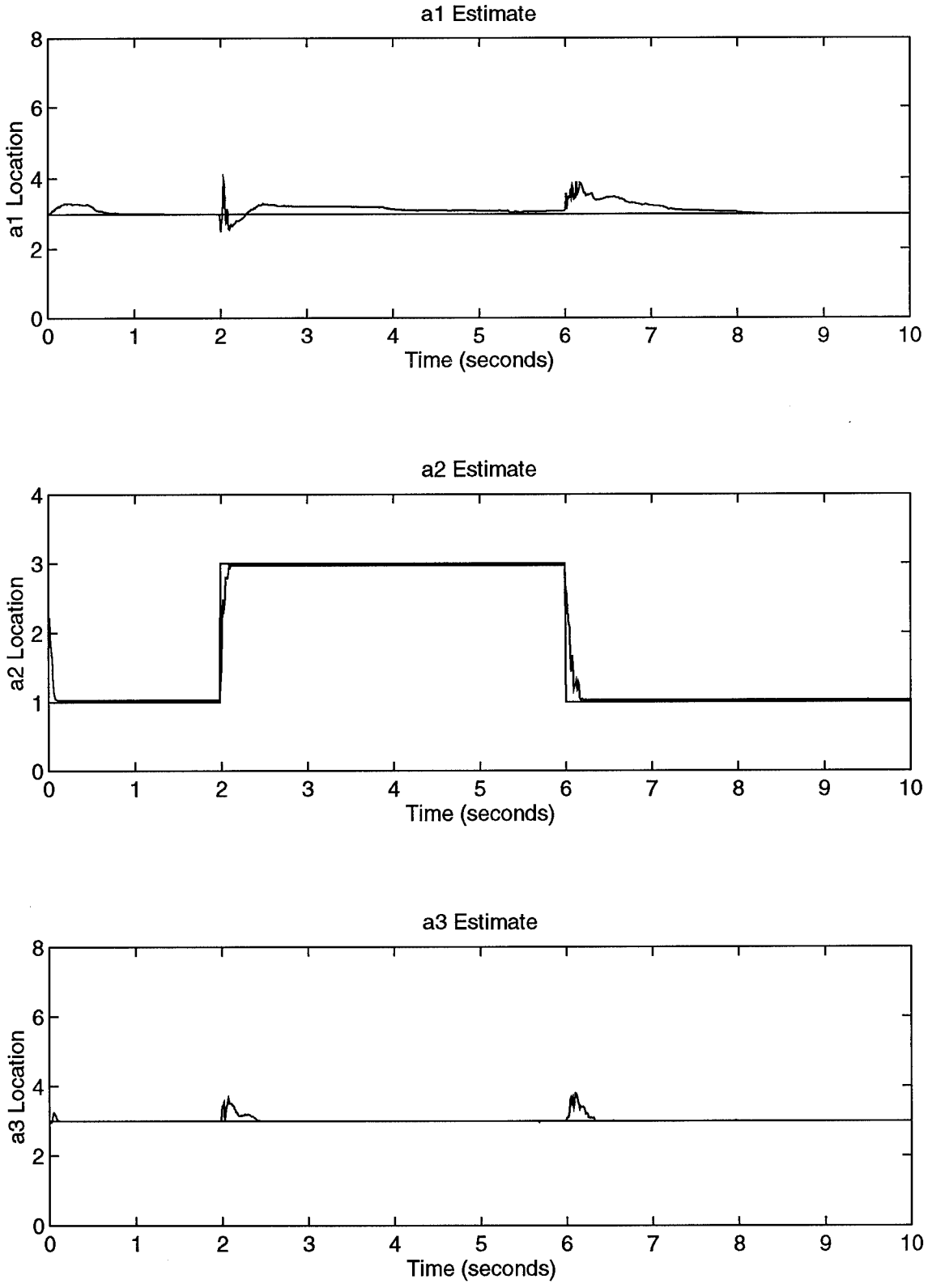


Figure A-76. a_2 Parameter Jumps Using Parameter Position Monitoring

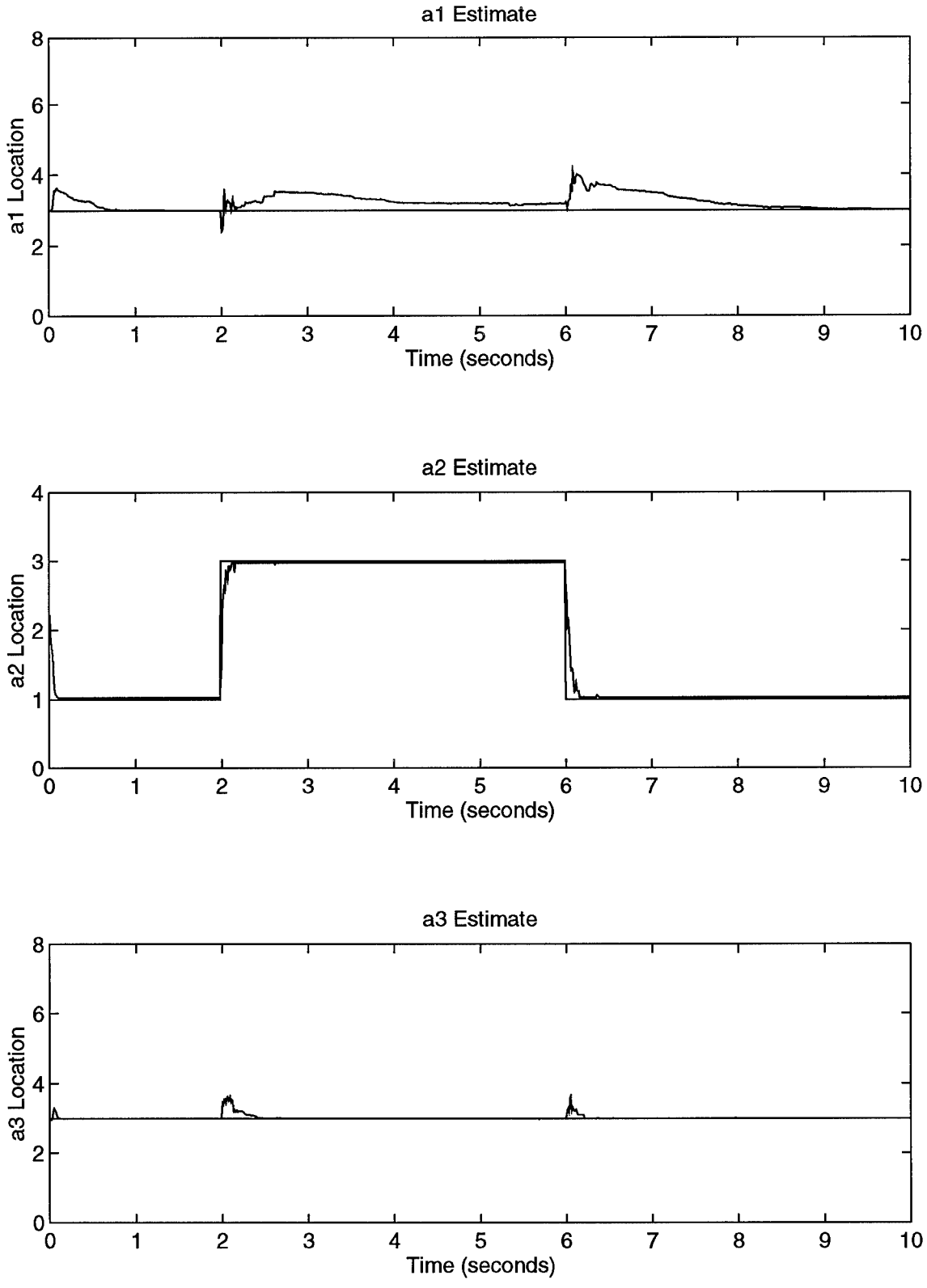


Figure A-77. a_2 Parameter Jumps Using Probability Monitoring

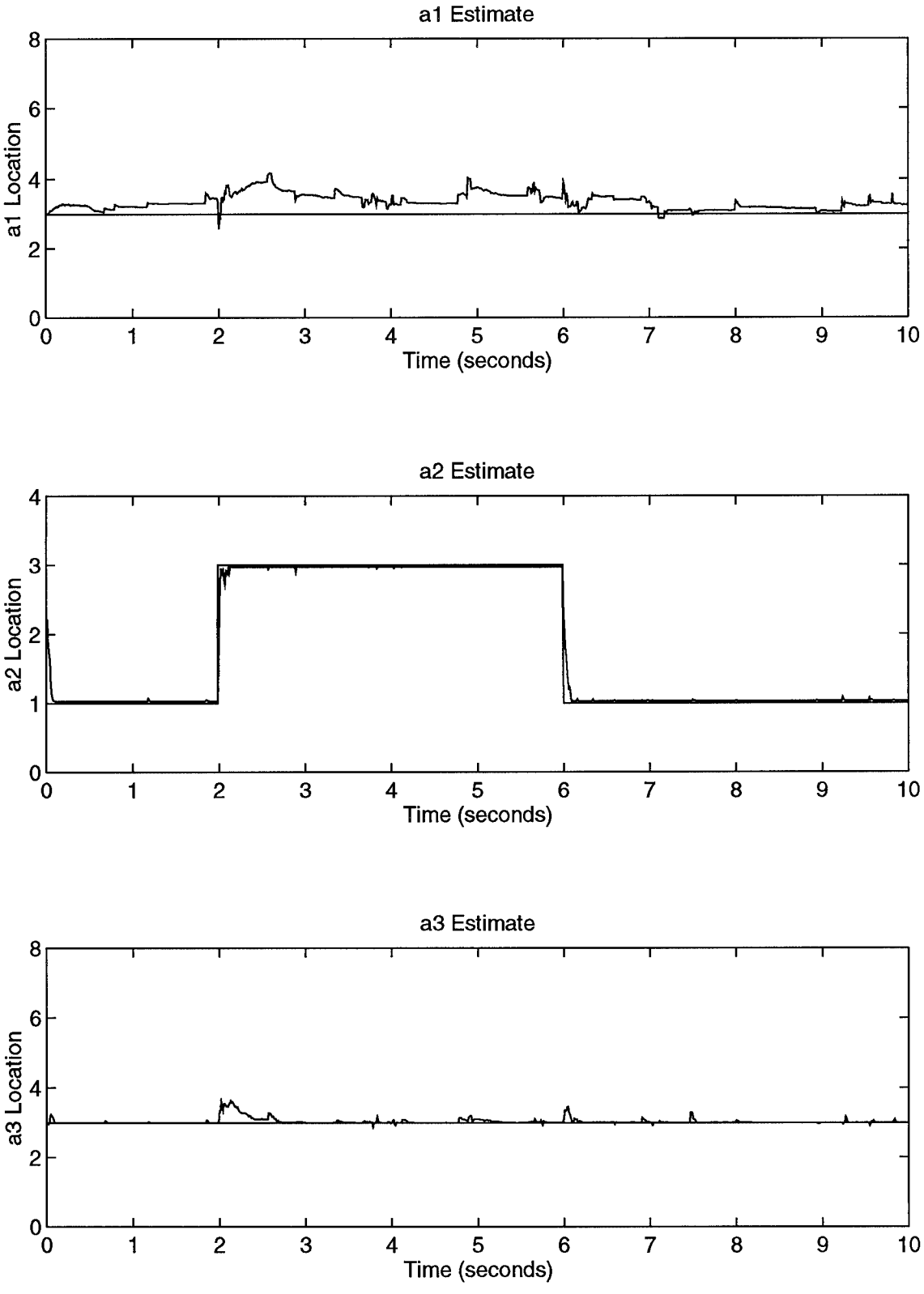


Figure A-78. a_2 Parameter Jumps Using Residual Monitoring

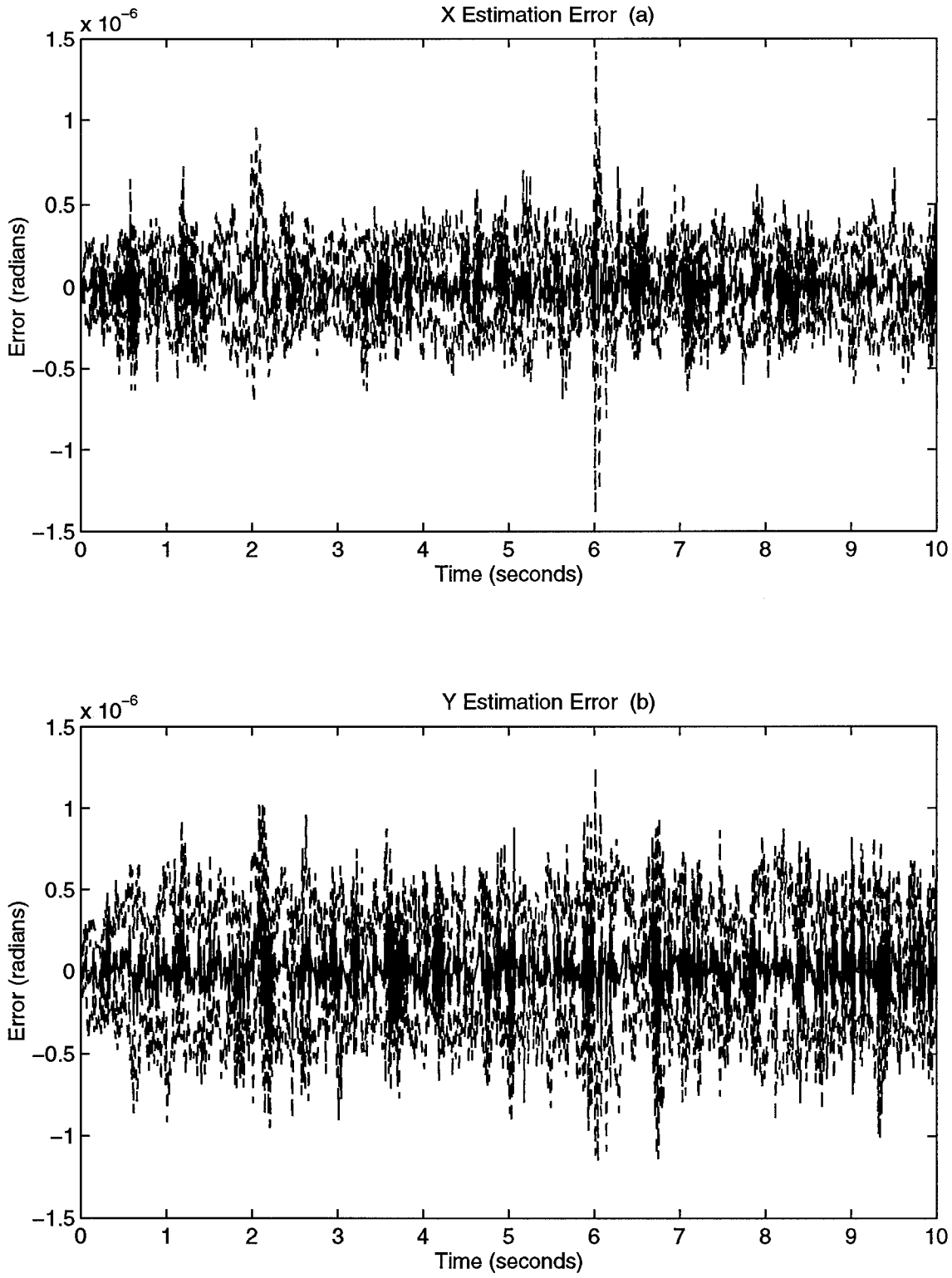


Figure A-79. a_2 Parameter Jumps Estimation Errors

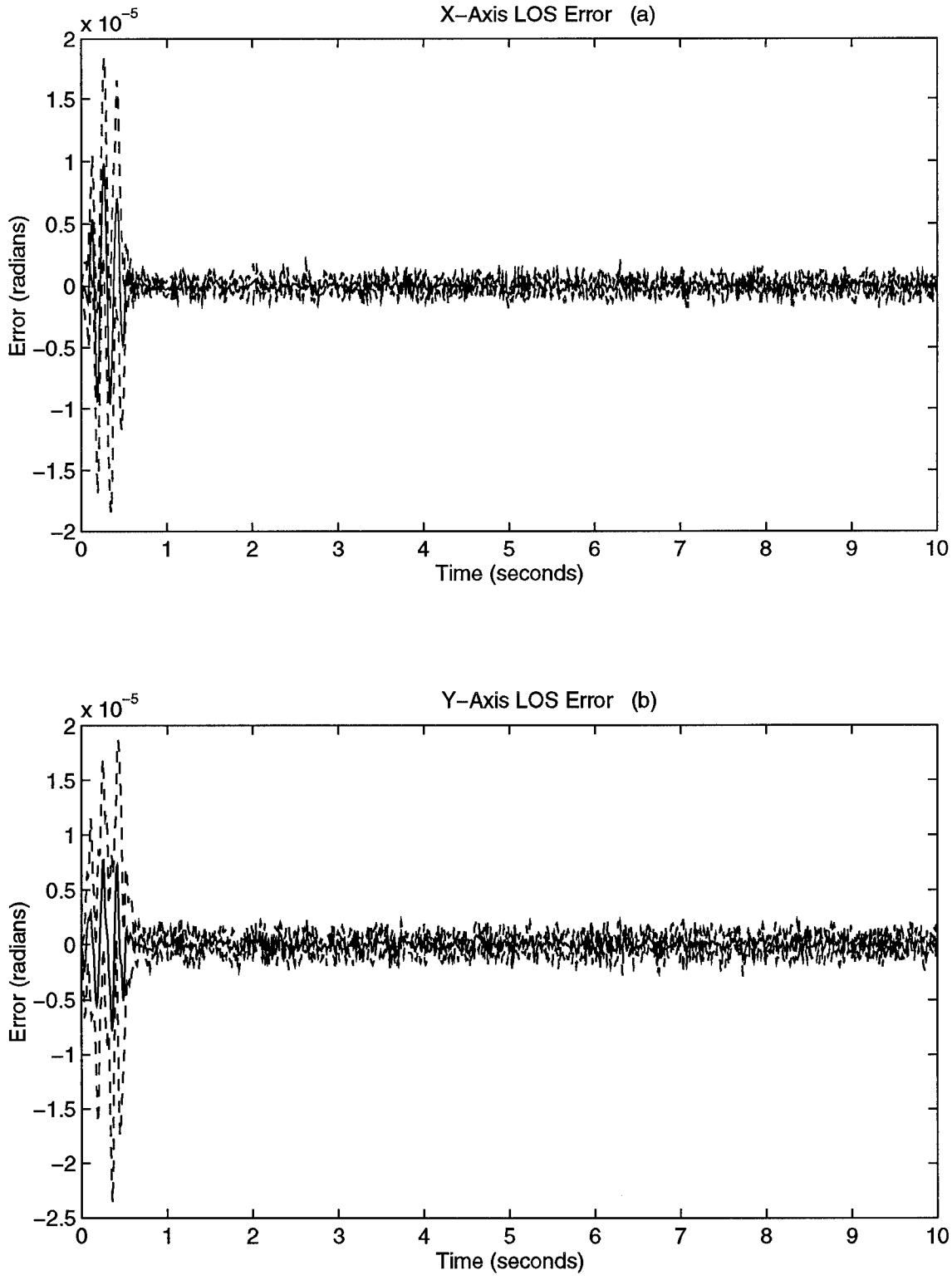


Figure A-80. a_2 Parameter Jumps LOS Errors

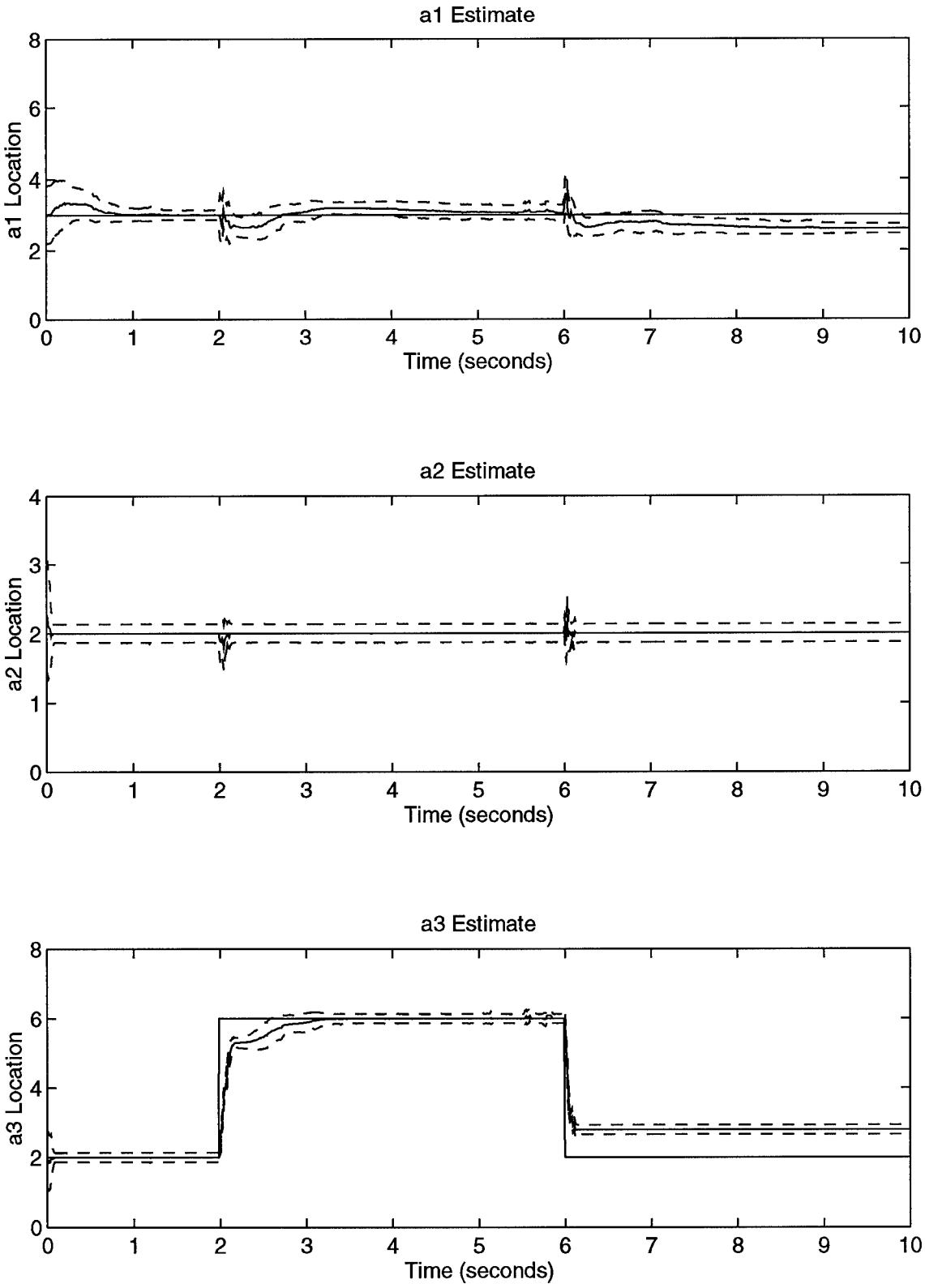


Figure A-81. a_3 Parameter Jumps Using Parameter Position Monitoring

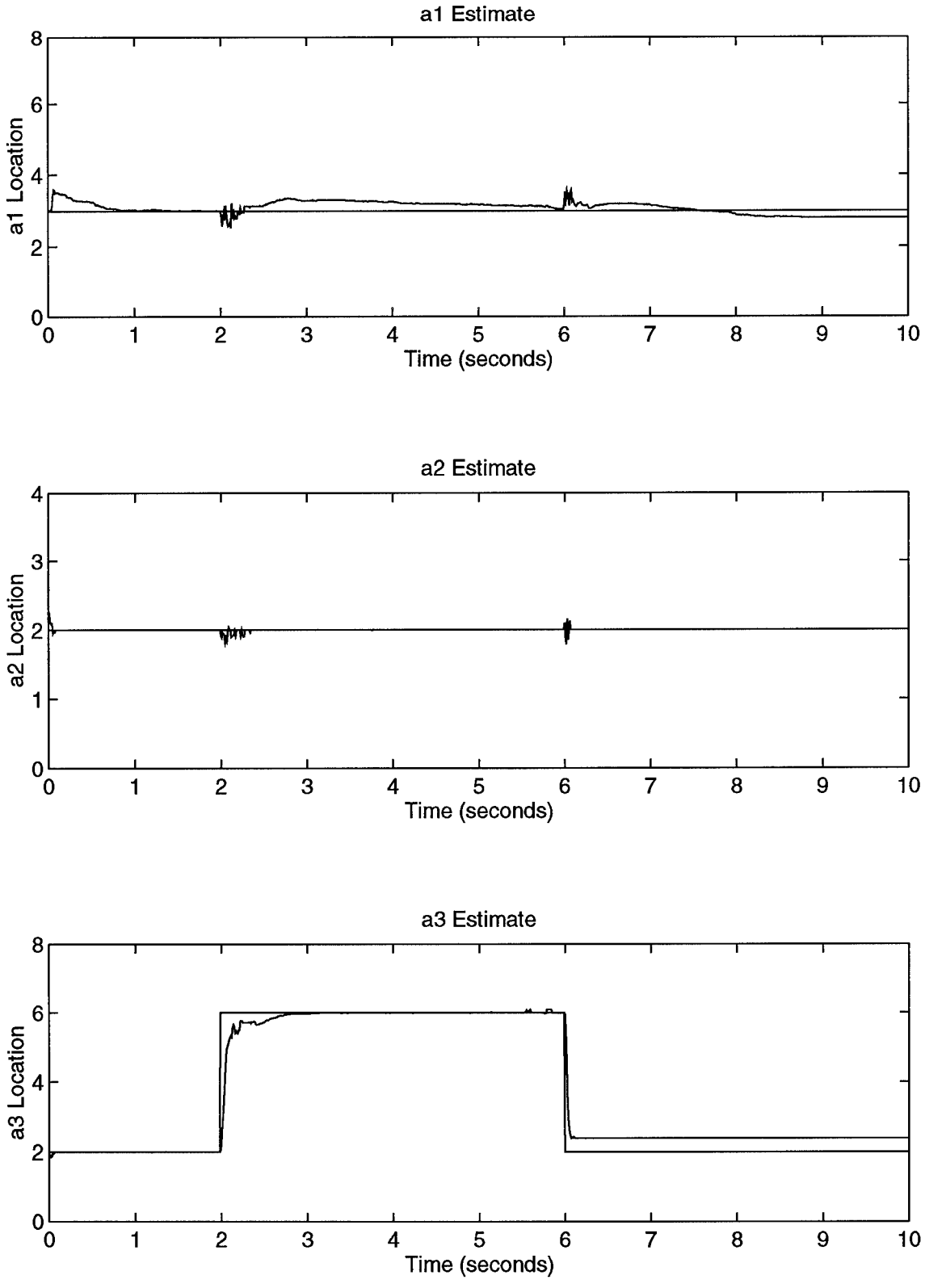


Figure A-82. a_3 Parameter Jumps Using Probability Monitoring

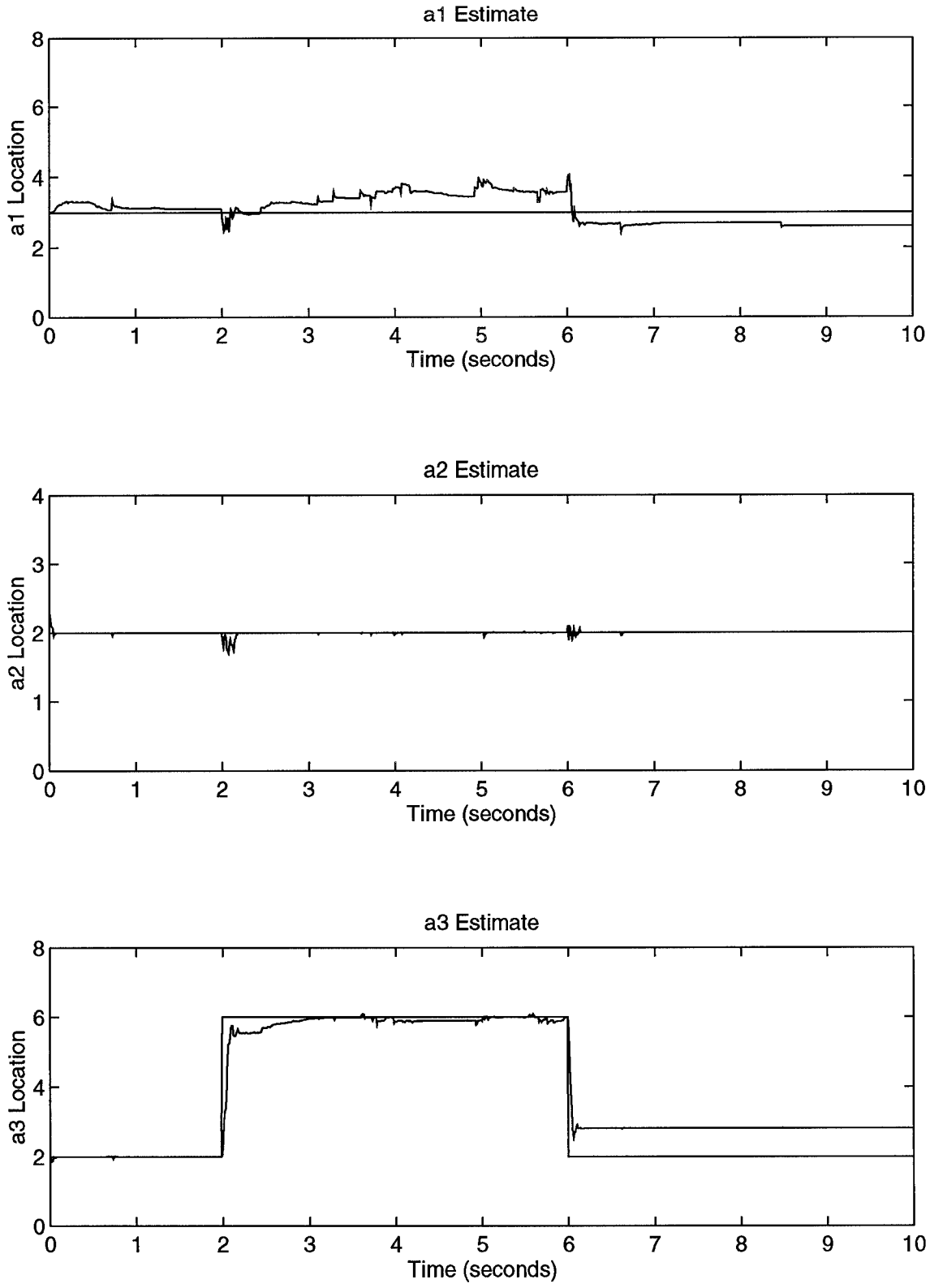


Figure A-83. a_3 Parameter Jumps Using Residual Monitoring

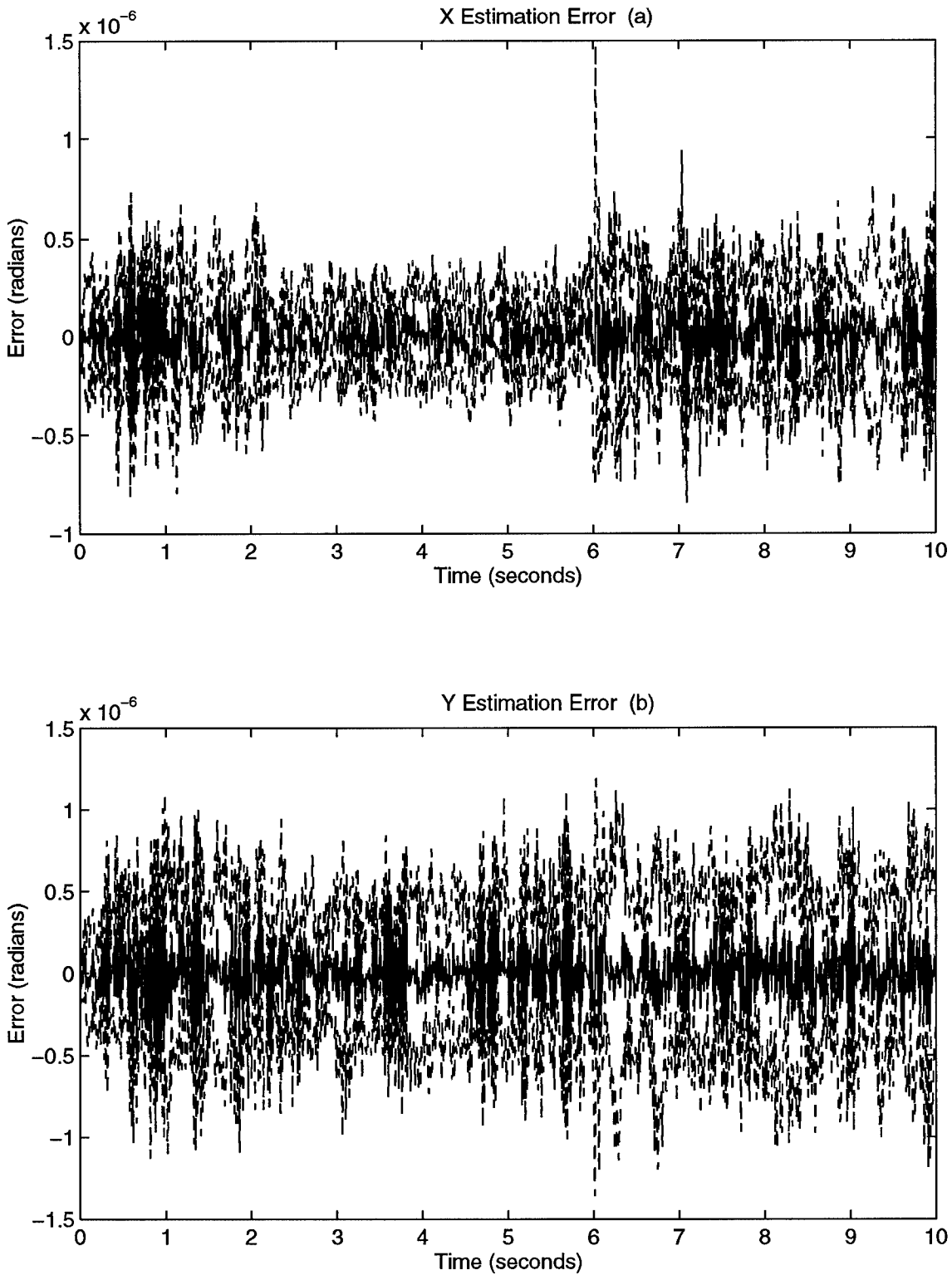


Figure A-84. a_3 Parameter Jumps Estimation Errors

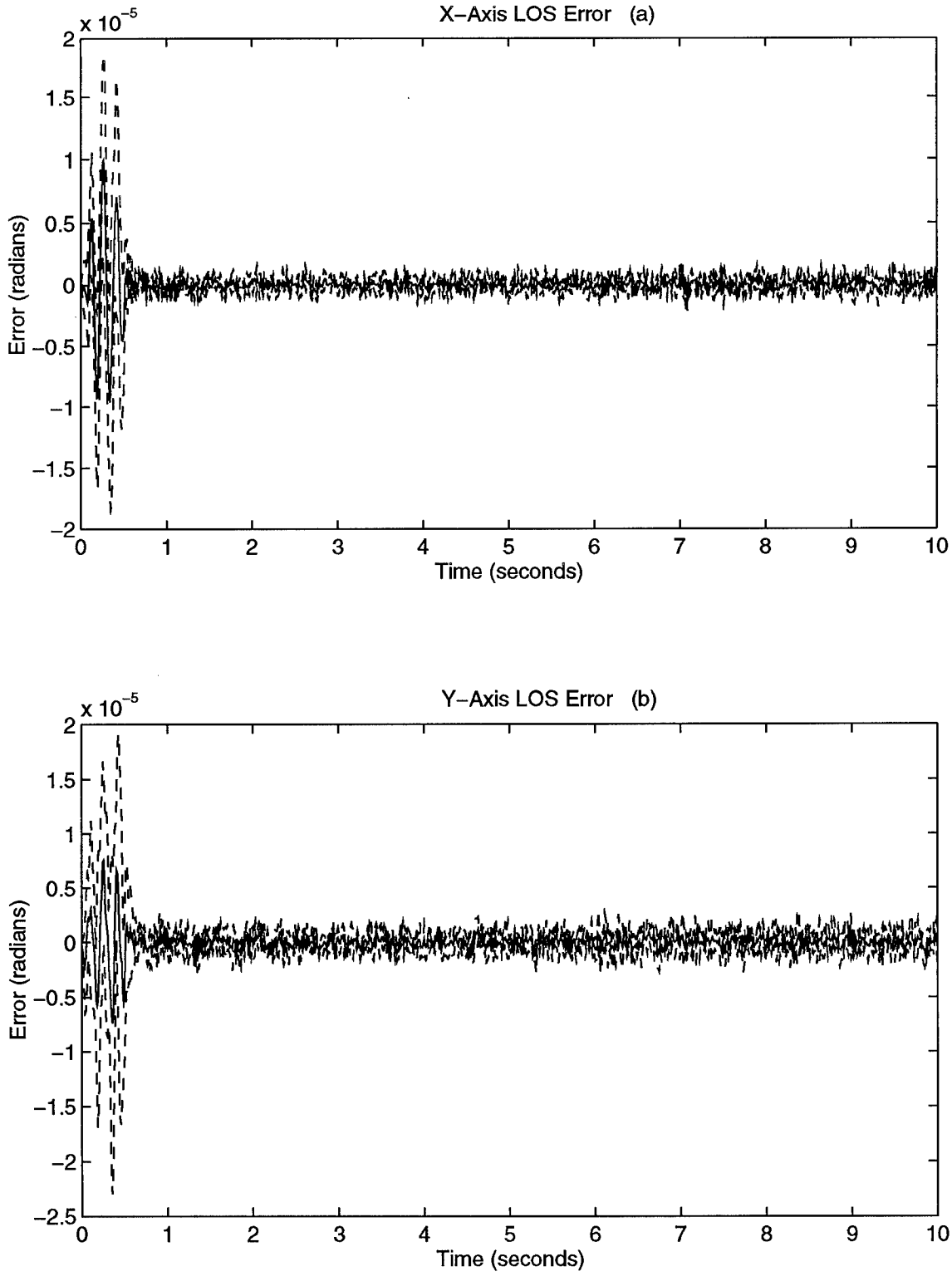


Figure A-85. a_3 Parameter Jumps LOS Errors

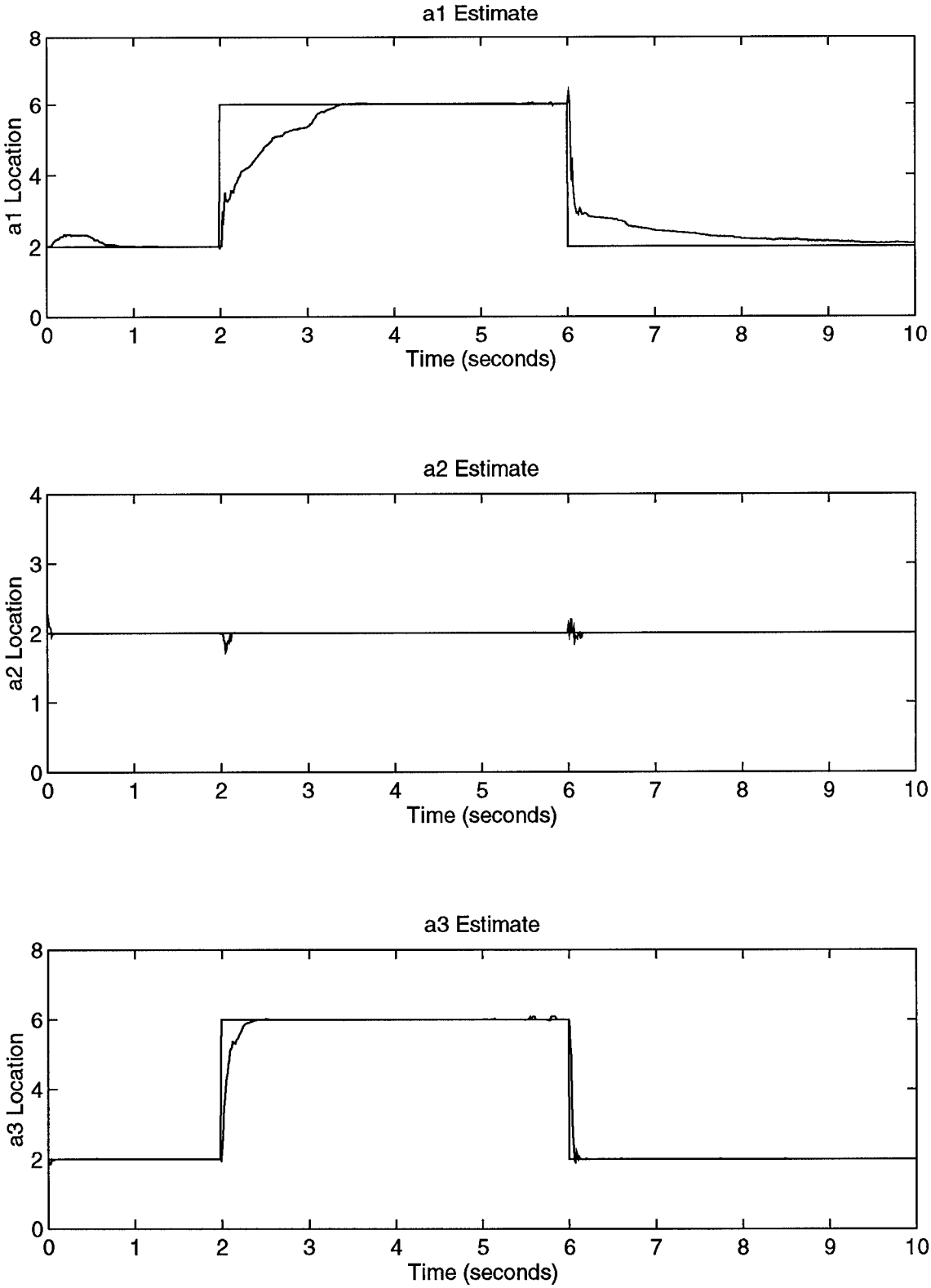


Figure A-86. a_1 and a_3 Jump Same Direction Using Parameter Position Monitoring

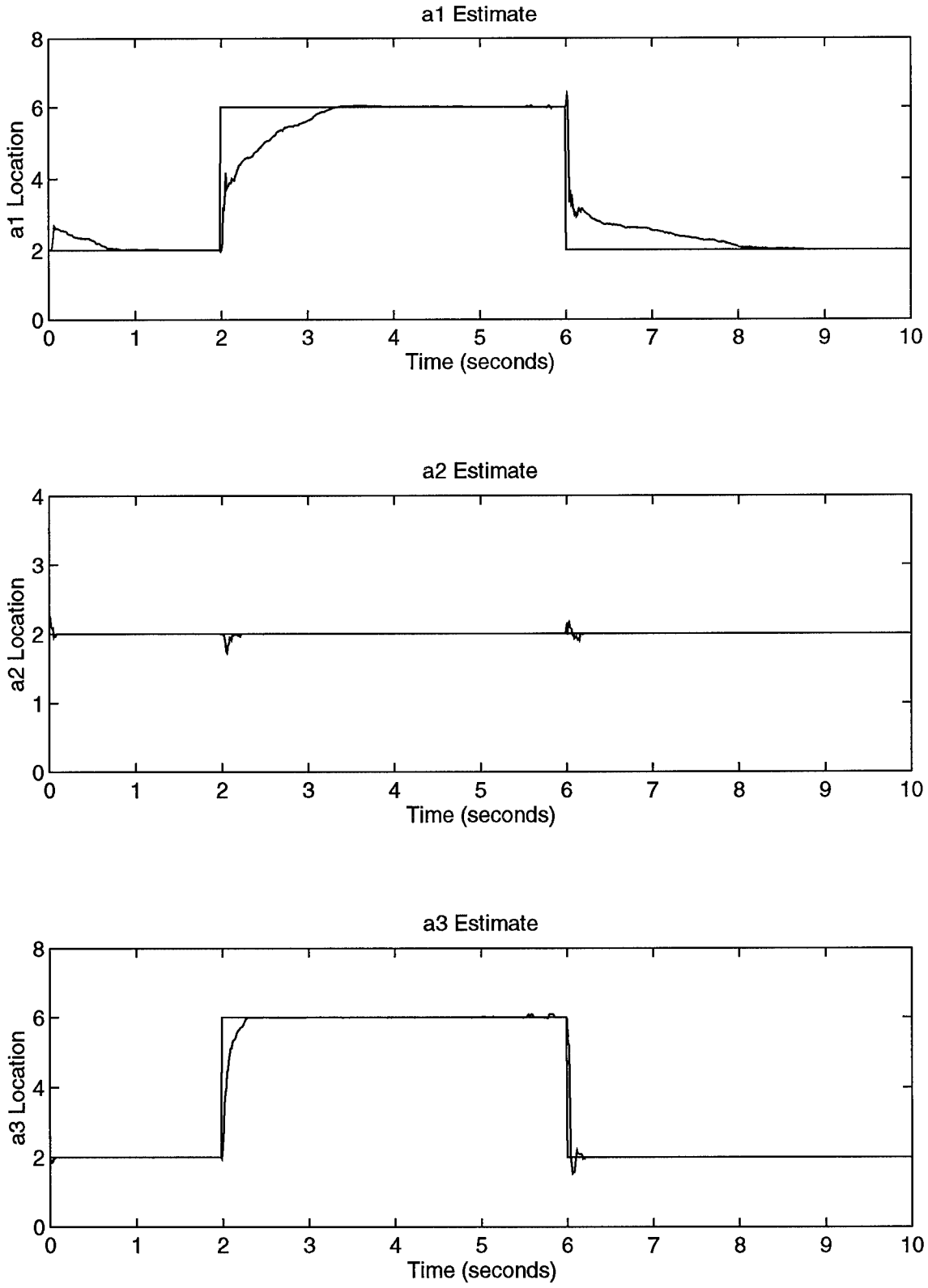


Figure A-87. a_1 and a_3 Jump Same Direction Using Probability Monitoring

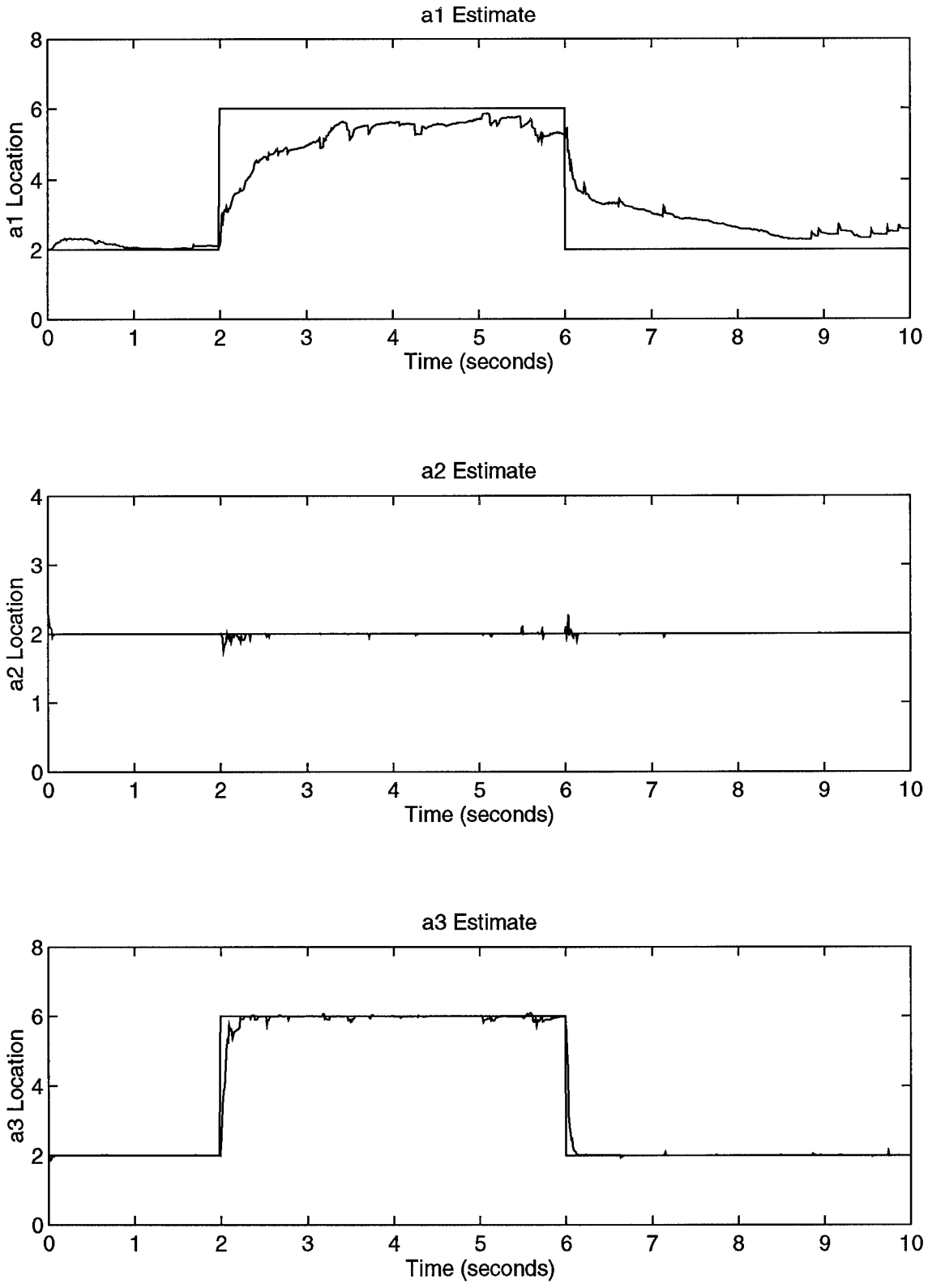
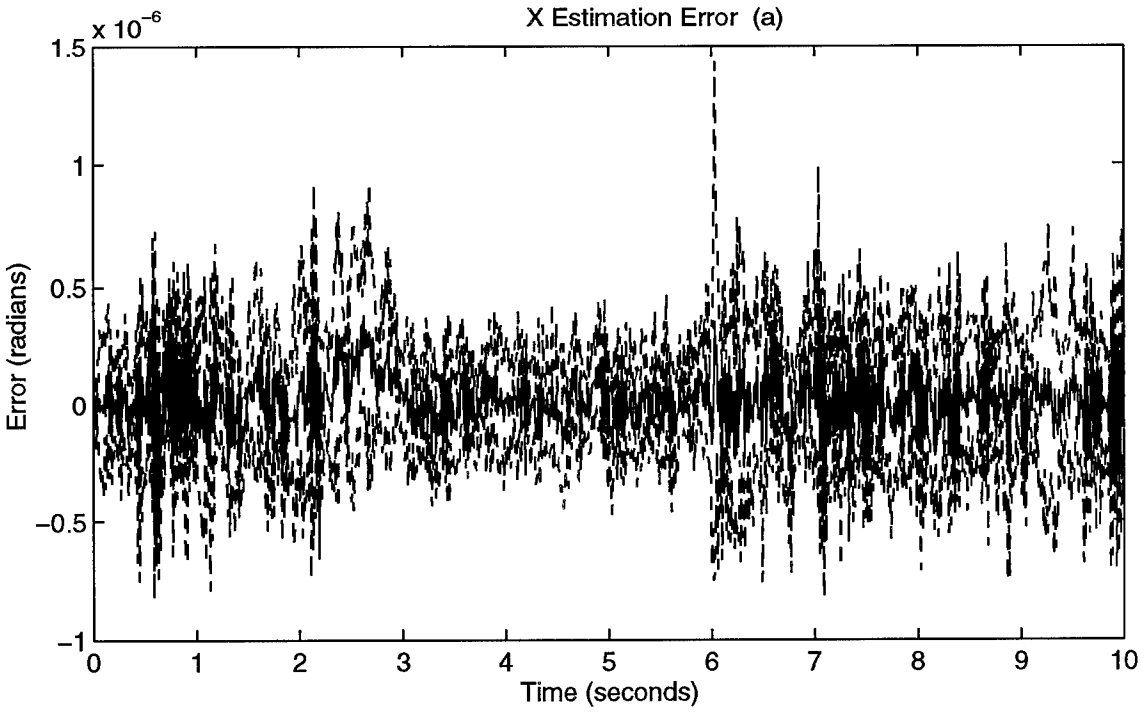


Figure A-88. a_1 and a_3 Jump Same Direction Using Residual Monitoring



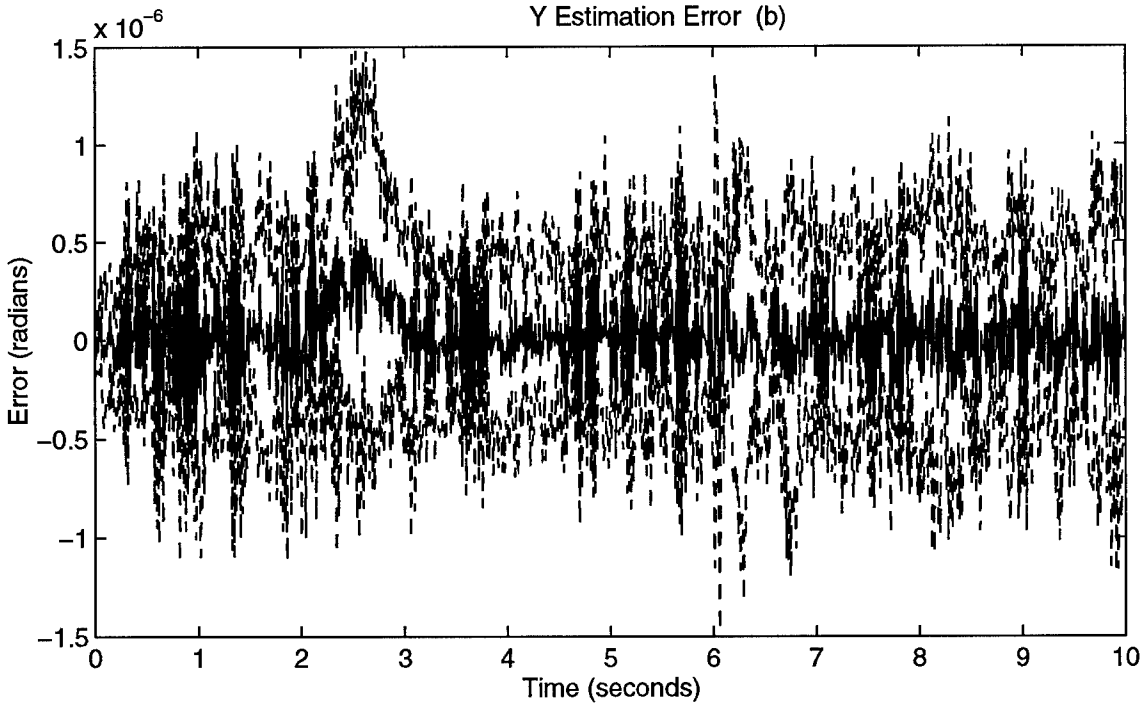
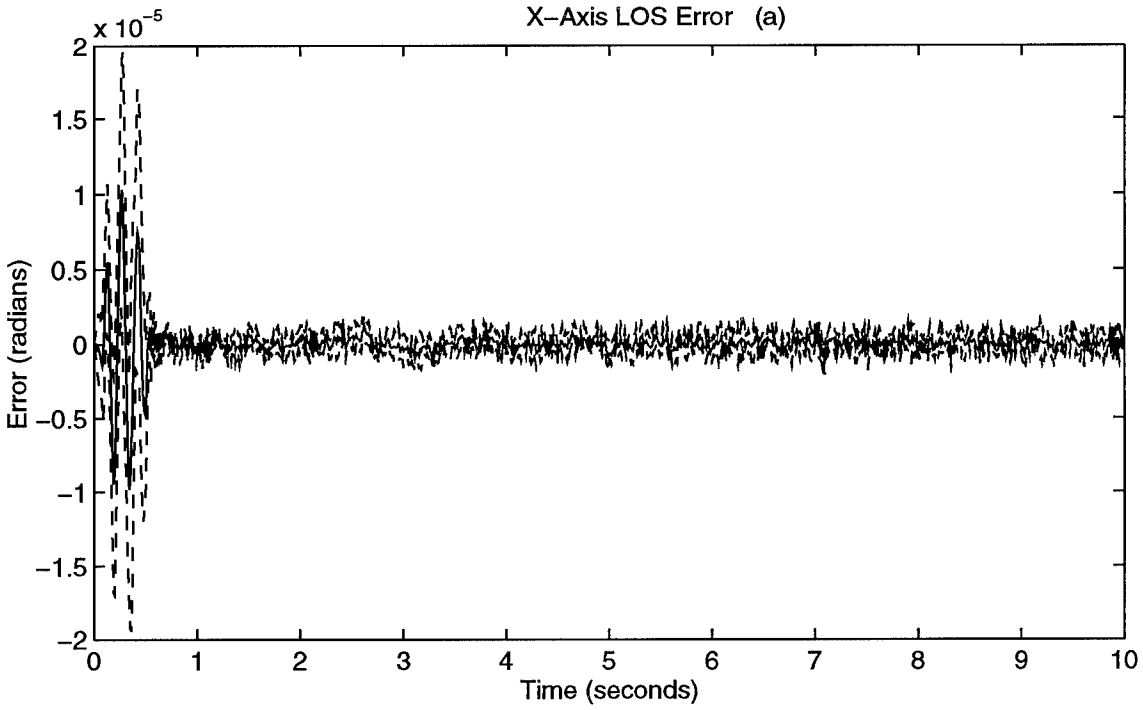


Figure A-89. a_1 and a_3 Jump Same Direction Estimation Errors



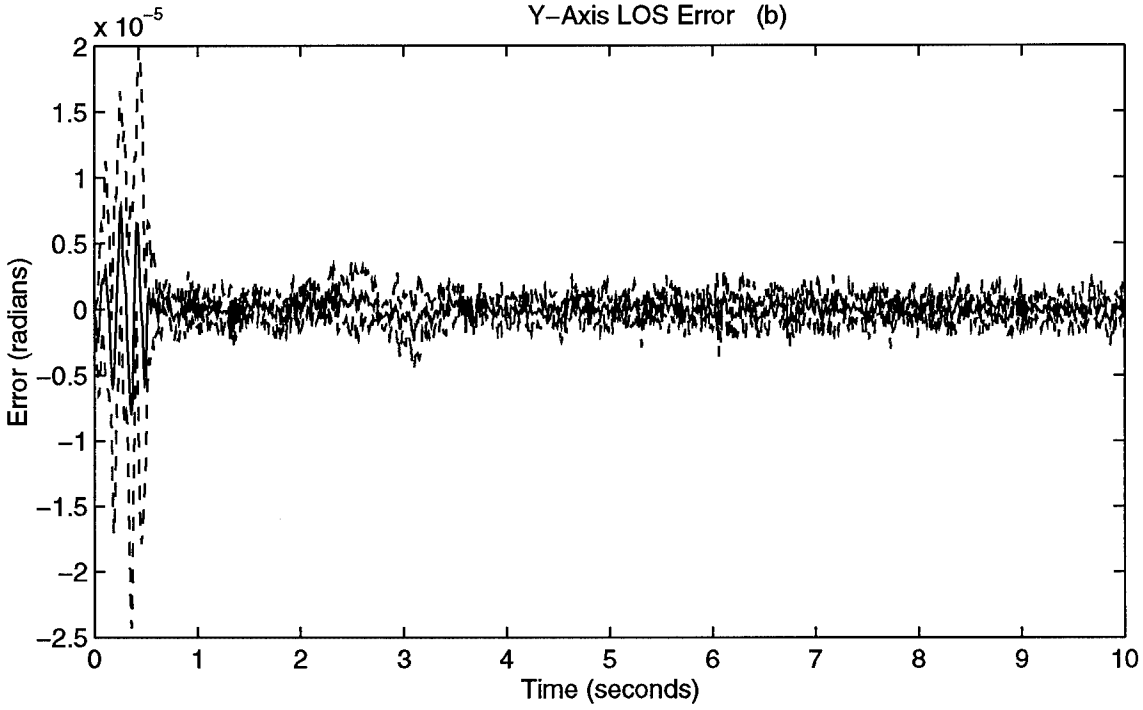


Figure A-90. a_1 and a_3 Jump Same Direction LOS Errors

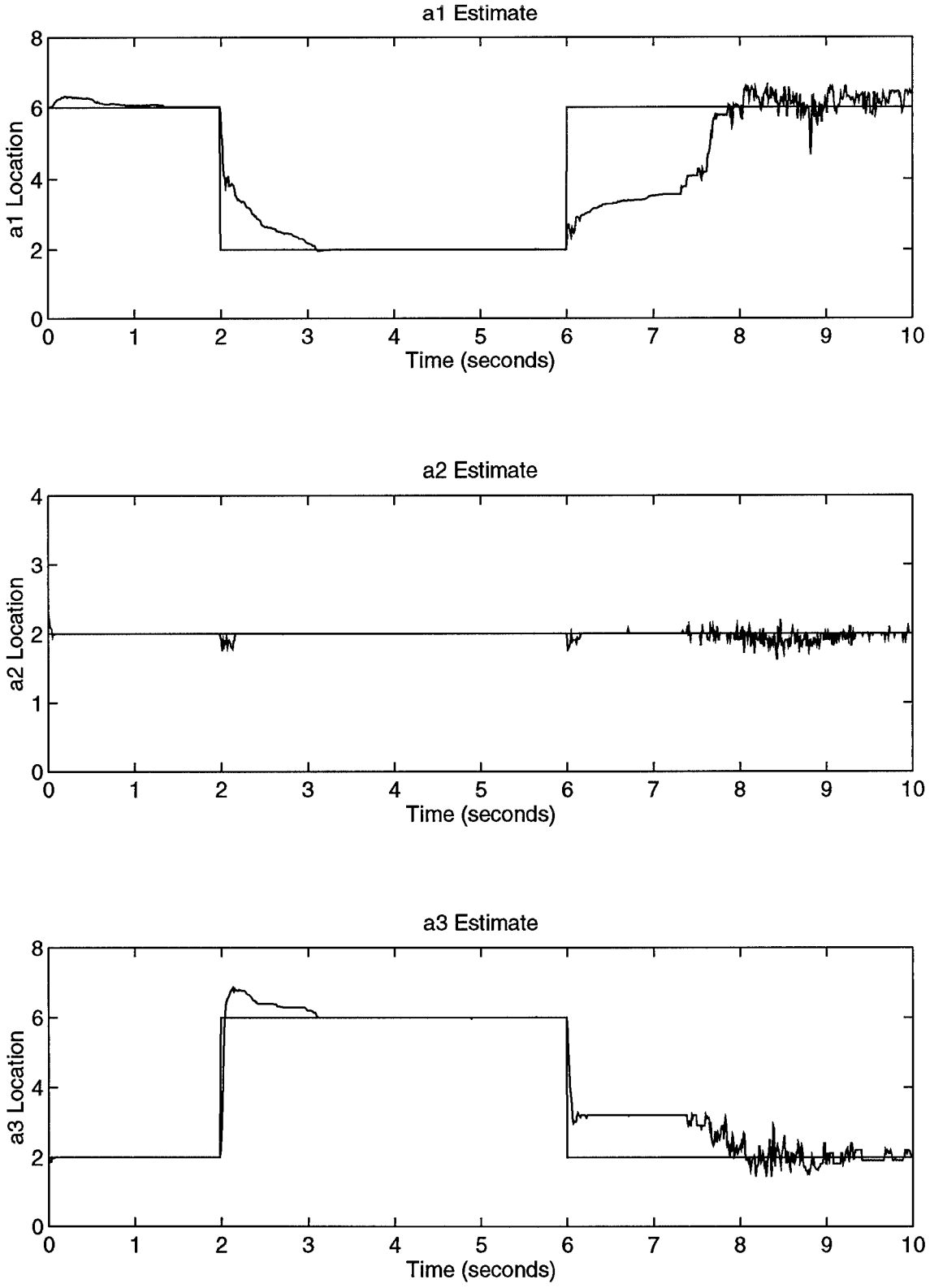


Figure A-91. a_1 and a_3 Jump Opposite Direction Using Parameter Position Monitoring

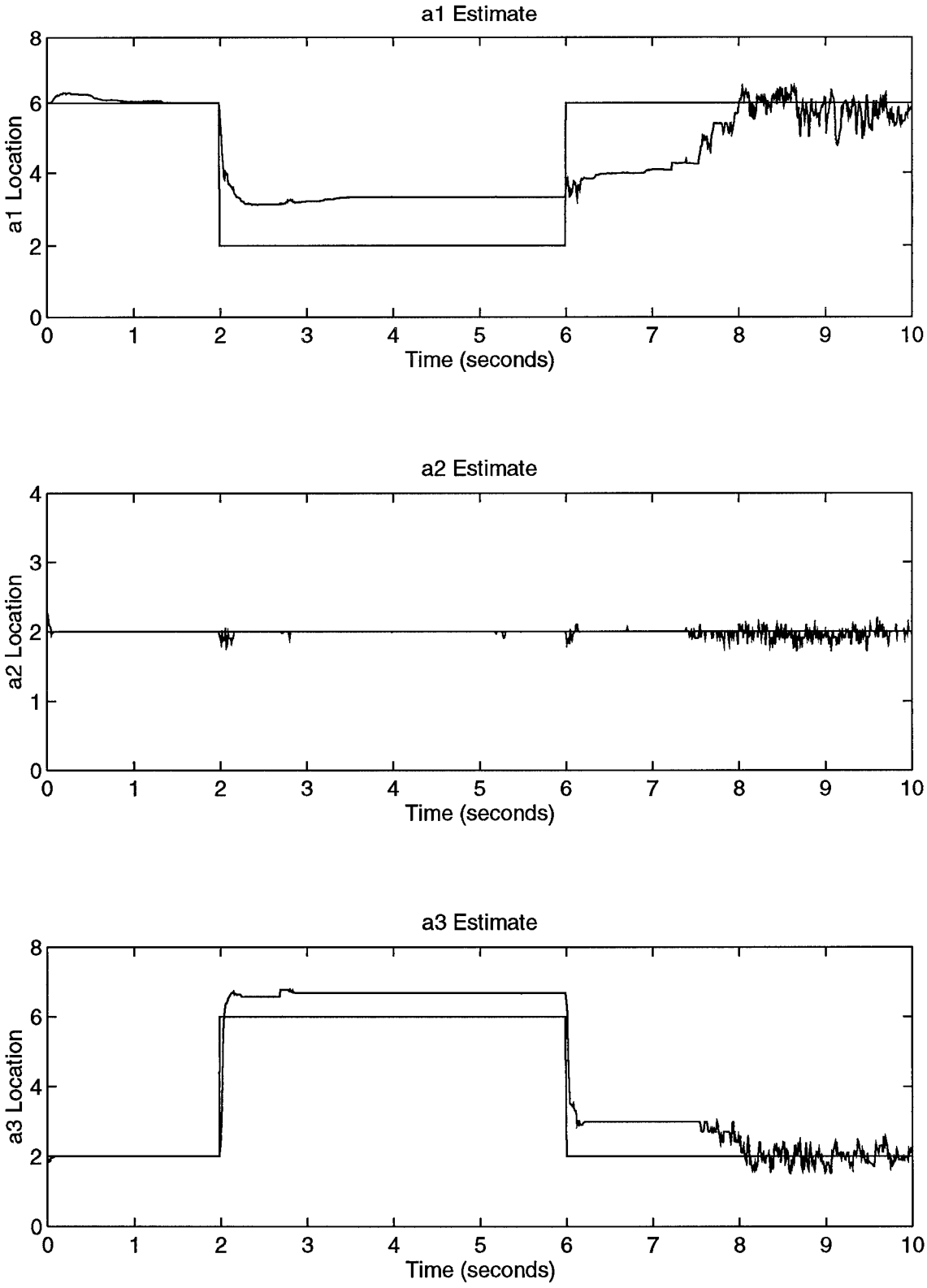


Figure A-92. a_1 and a_3 Jump Opposite Direction Using Probability Monitoring

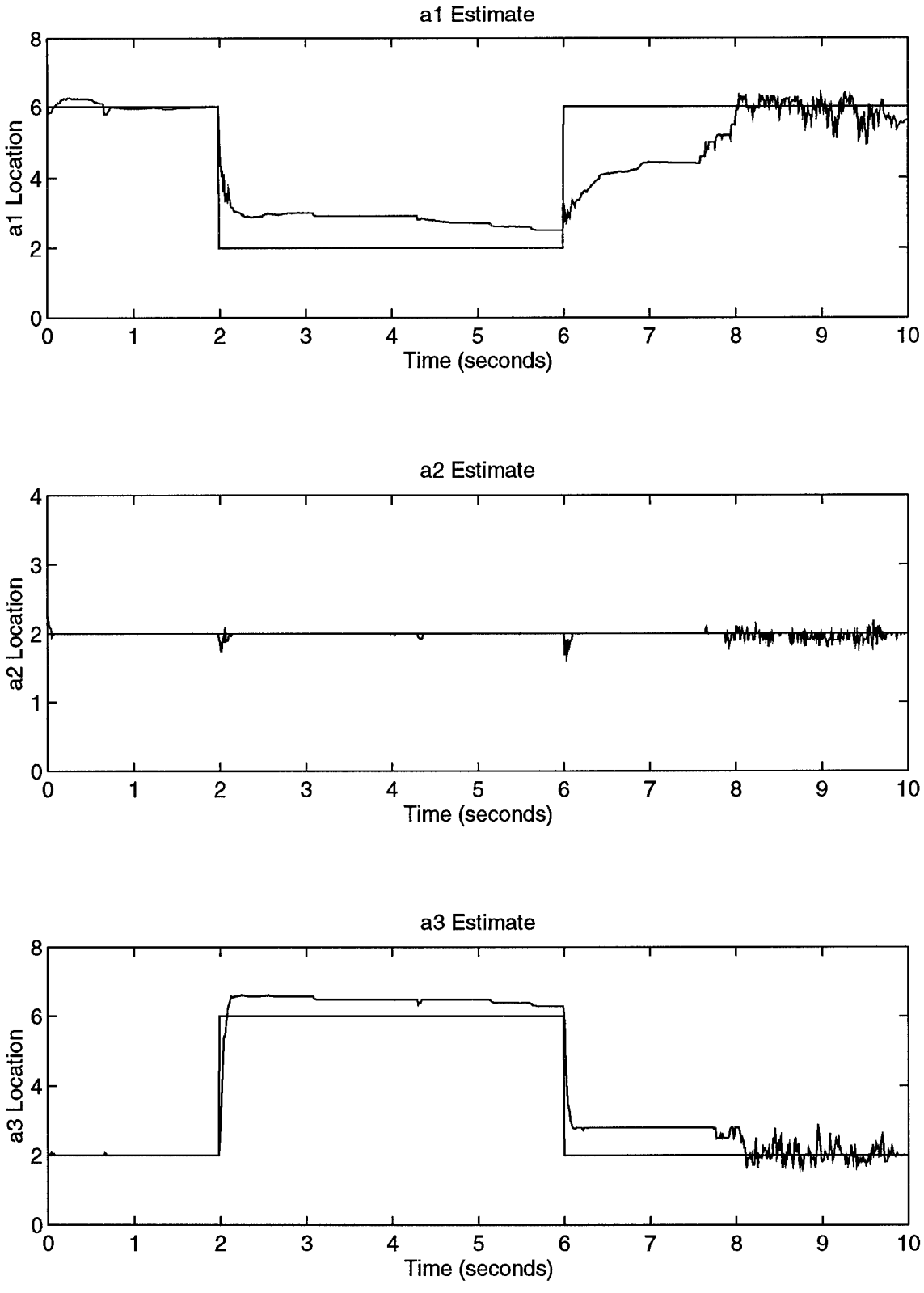


Figure A-93. a_1 and a_3 Jump Opposite Direction Using Residual Monitoring

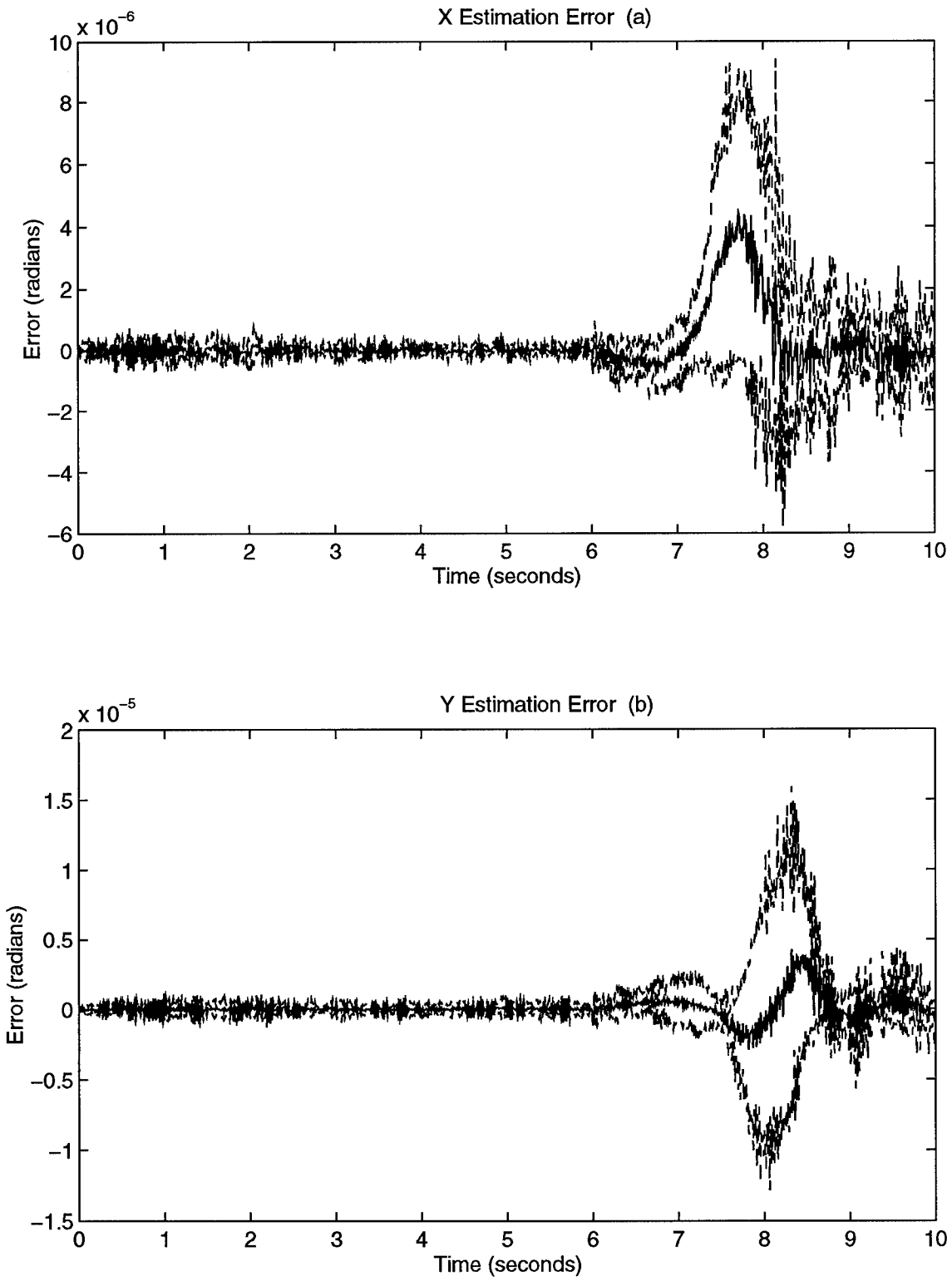


Figure A-94. a_1 and a_3 Jump Opposite Direction Estimation Errors

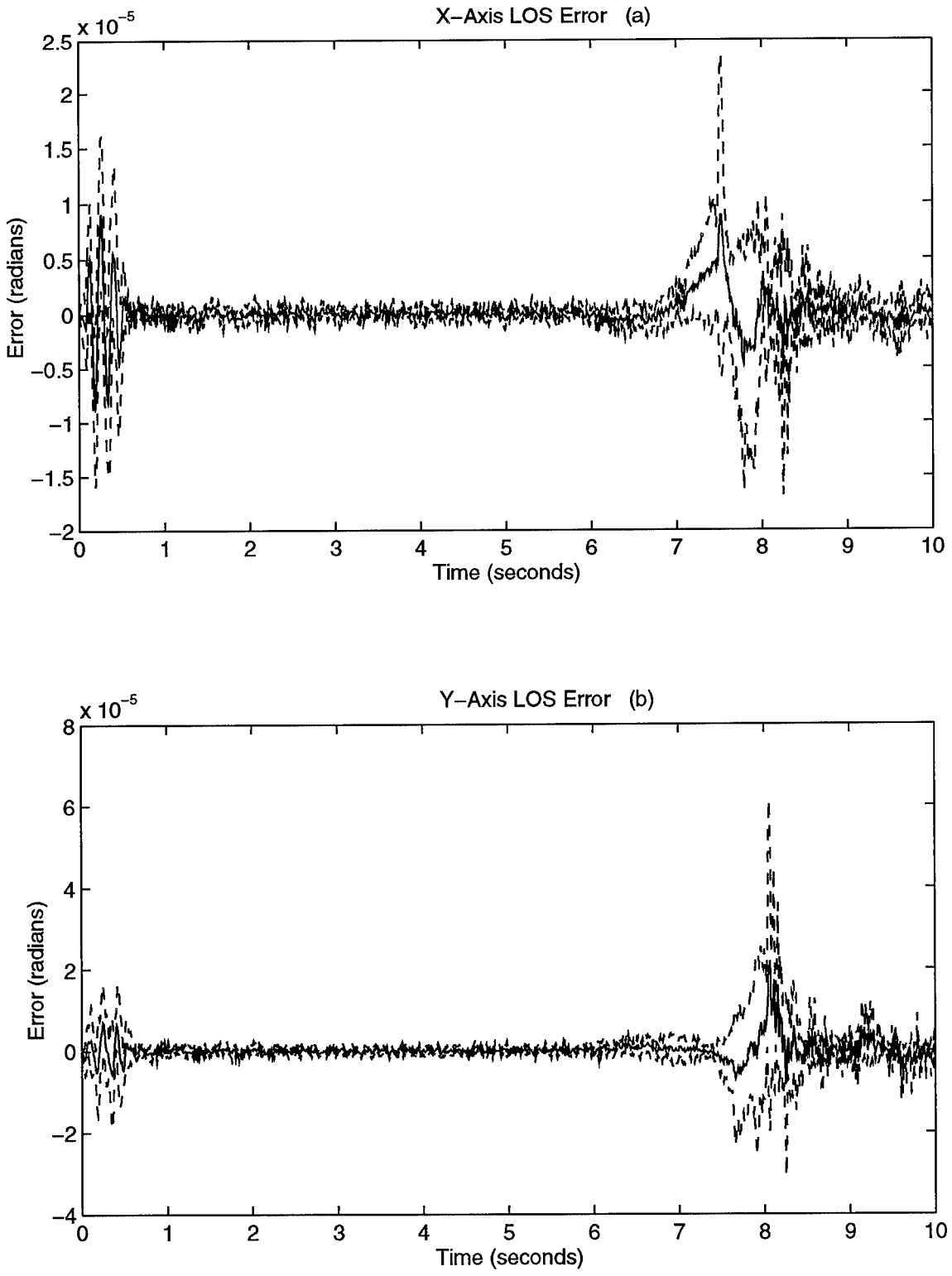


Figure A-95. a_1 and a_3 Jump Opposite Direction LOS Errors

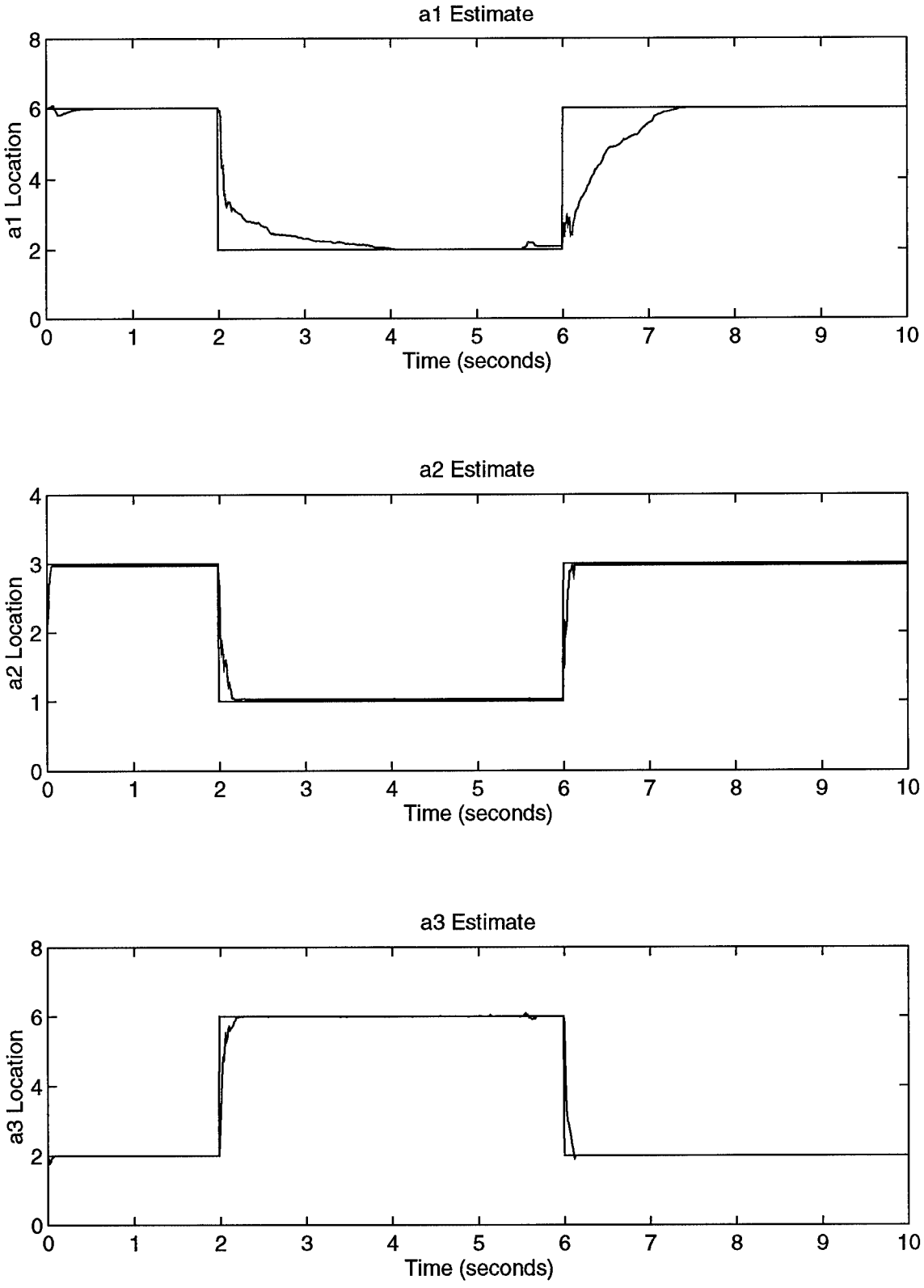


Figure A-96. 3 Parameter Jump (a_1 and a_3 Opposite) Using Parameter Position Mon.

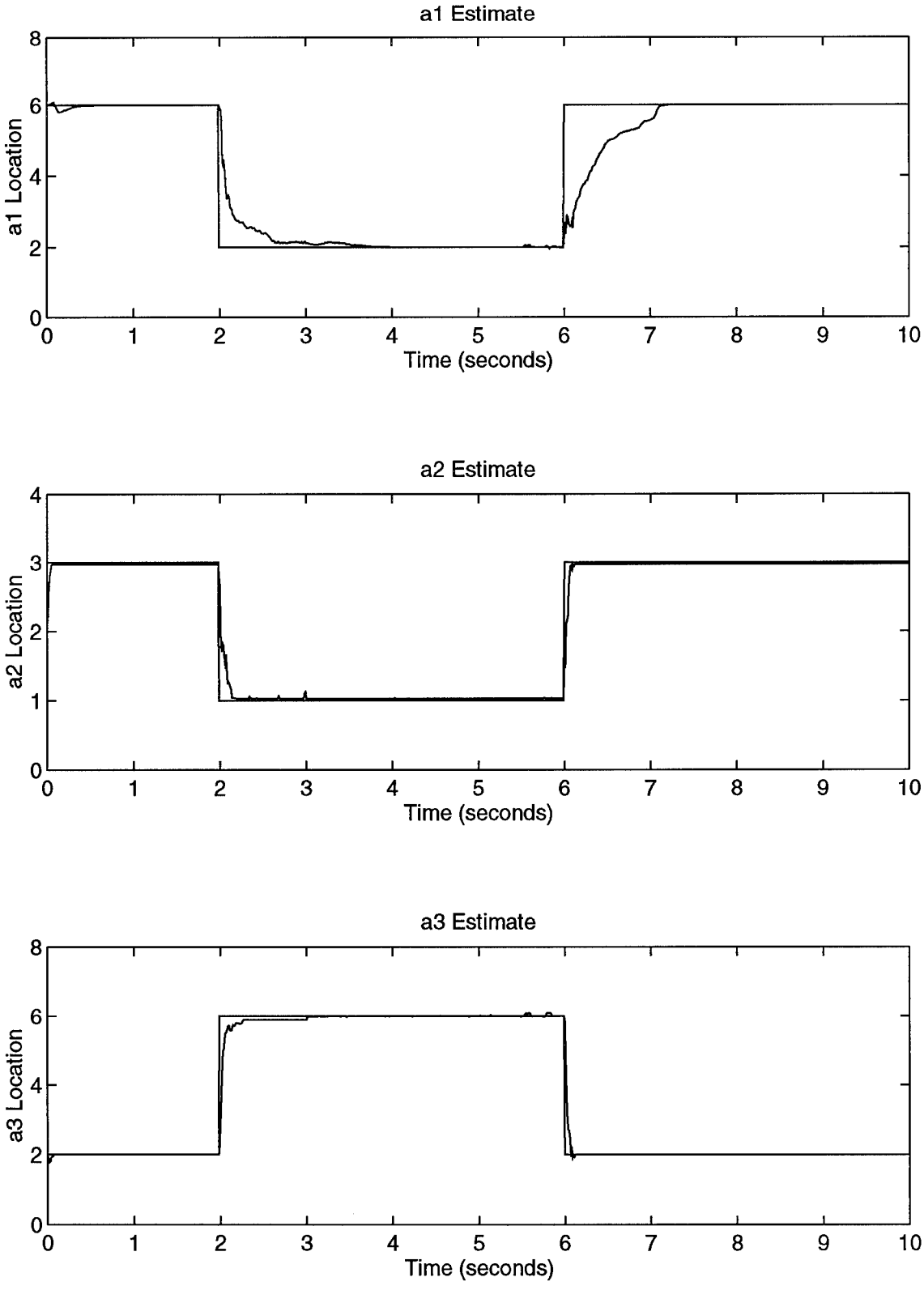


Figure A-97. 3 Parameter Jump (a_1 and a_3 Opposite) Using Probability Monitoring

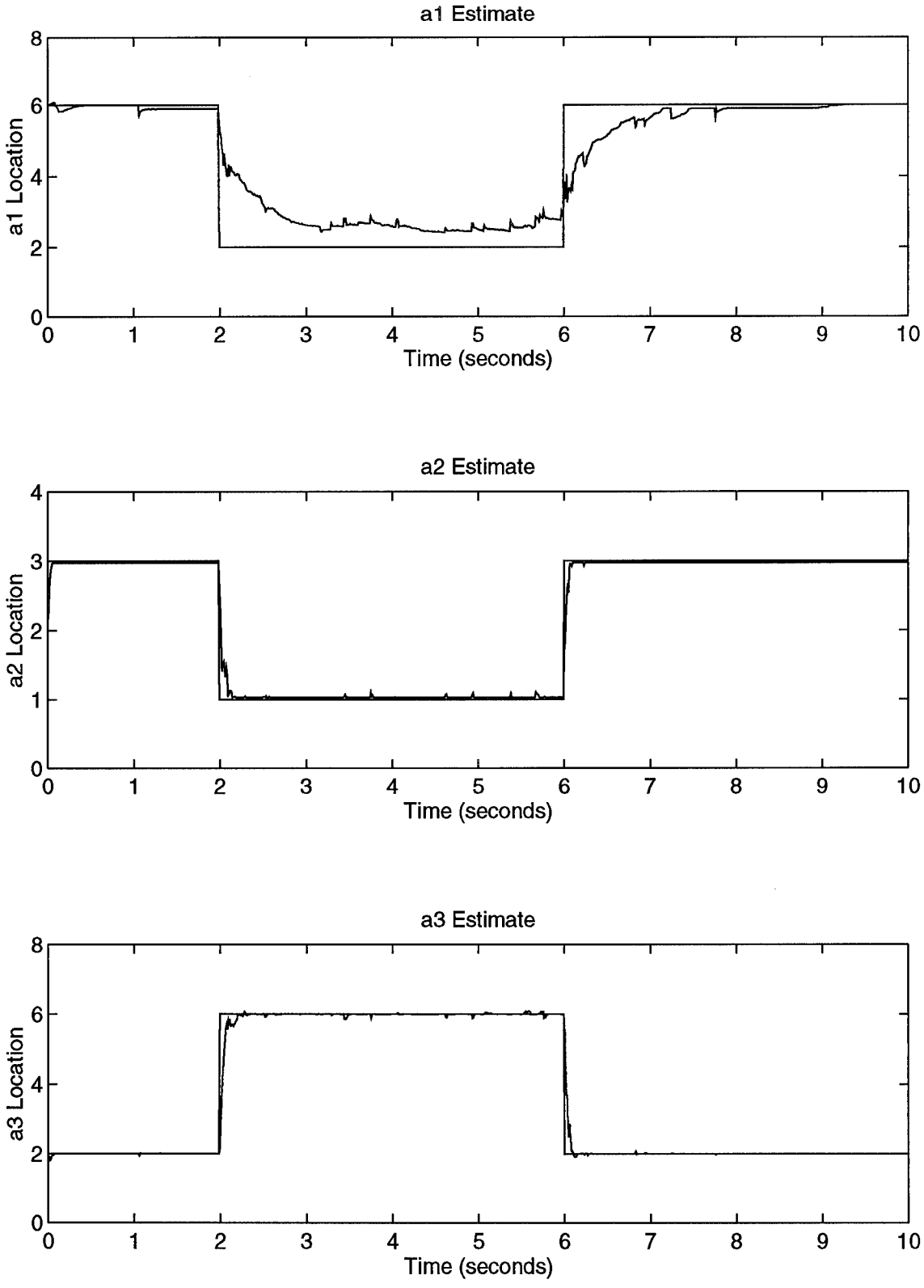
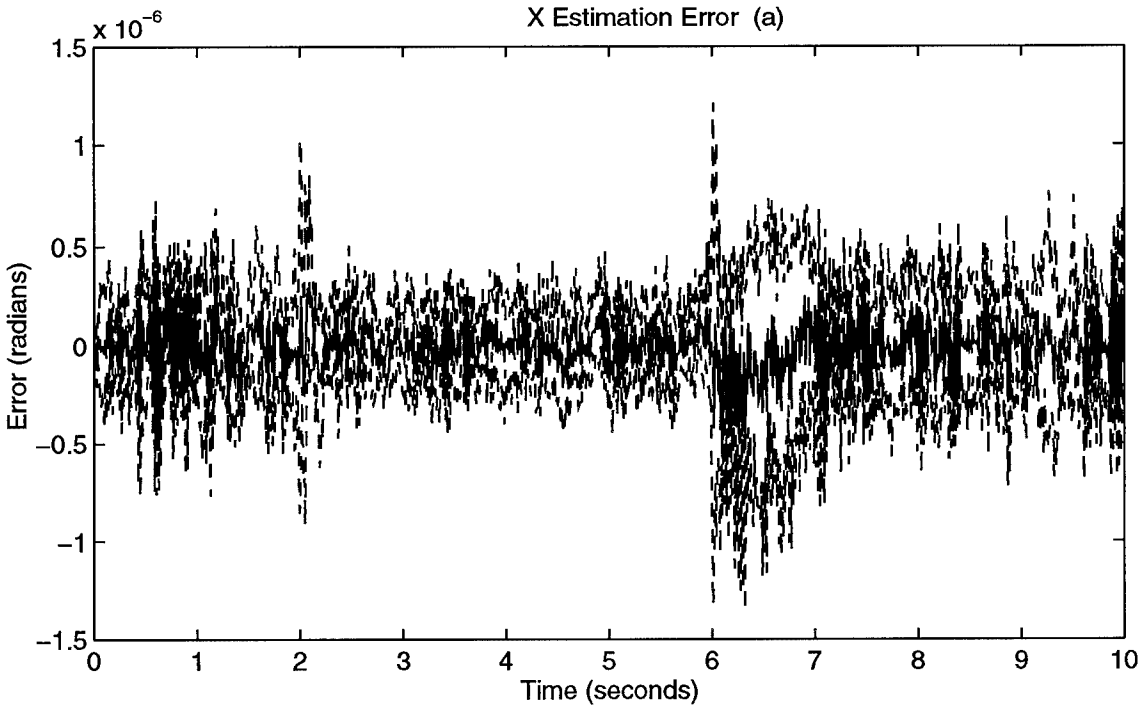


Figure A-98. 3 Parameter Jump (a_1 and a_3 Opposite) Using Residual Monitoring



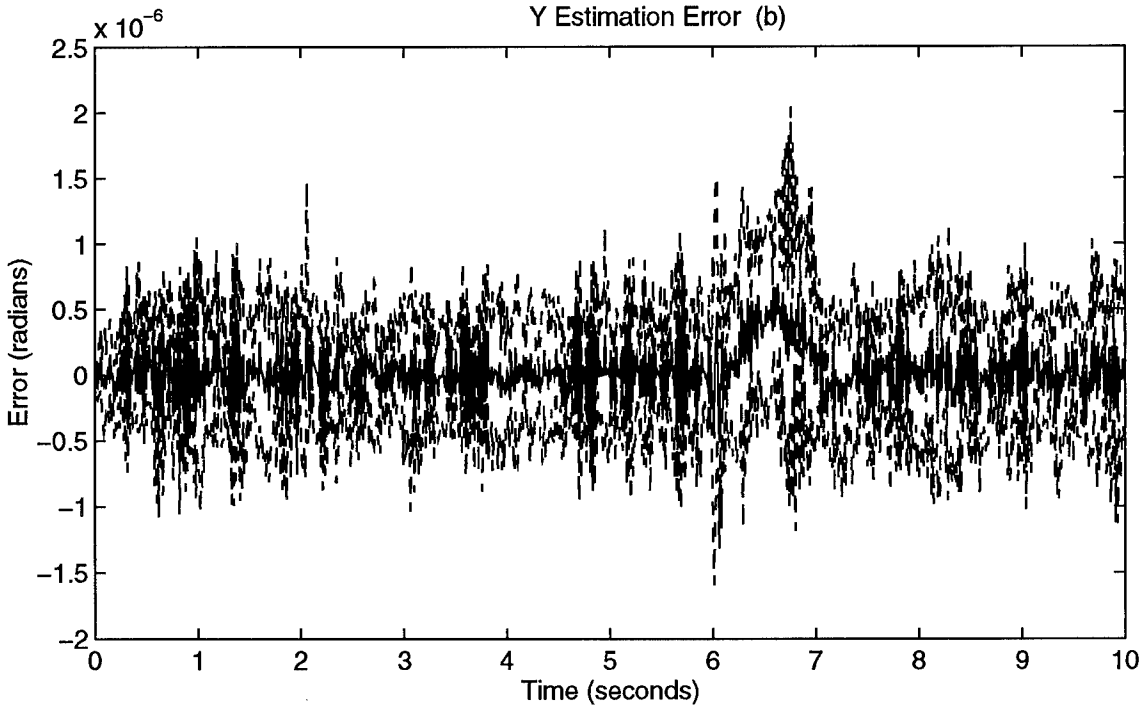
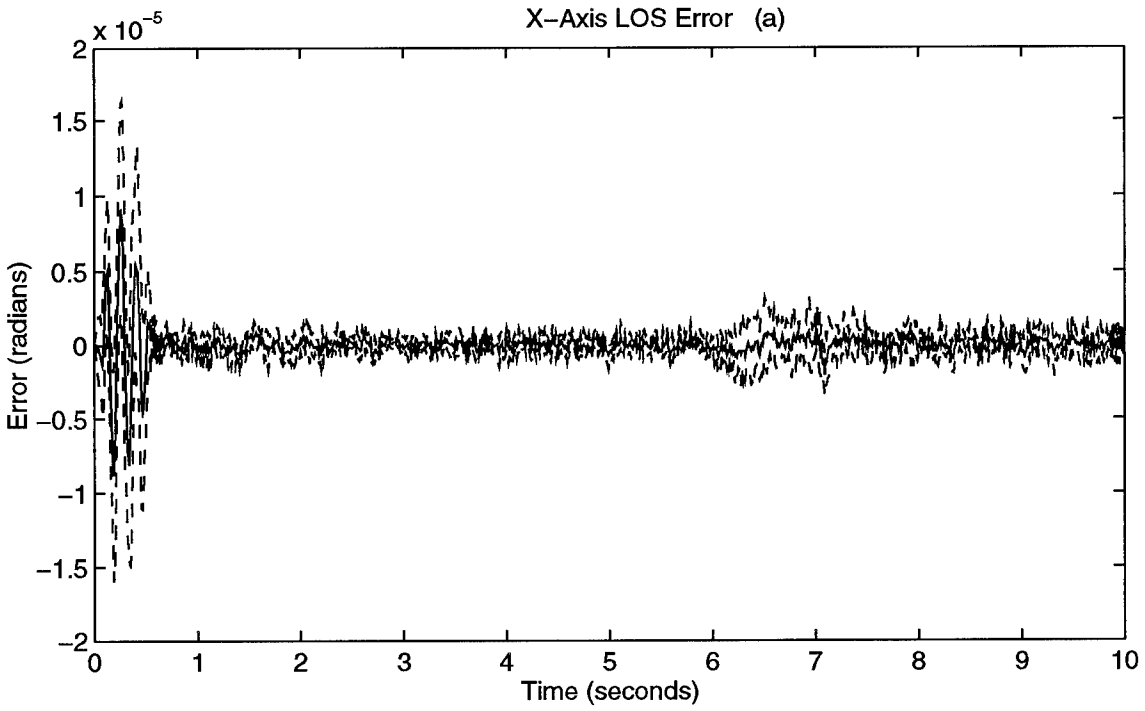


Figure A-99. 3 Parameter Jump (a_1 and a_3 Opposite) Estimation Errors



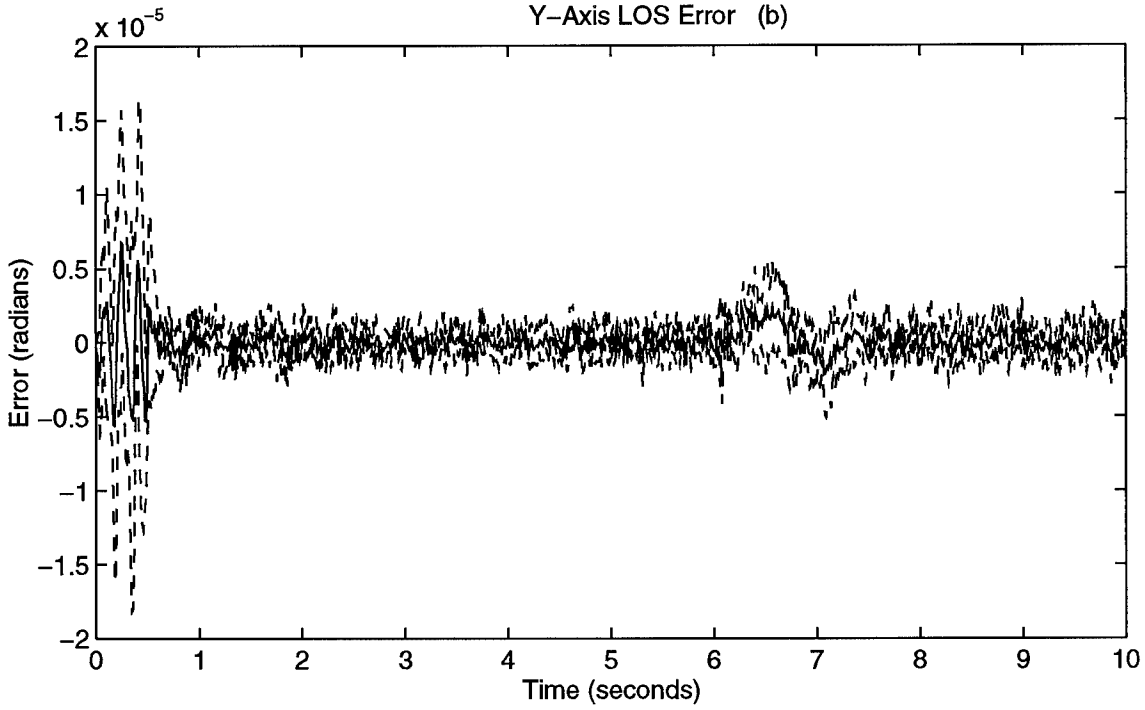


Figure A-100. 3 Parameter Jump (a_1 and a_3 Opposite) LOS Errors

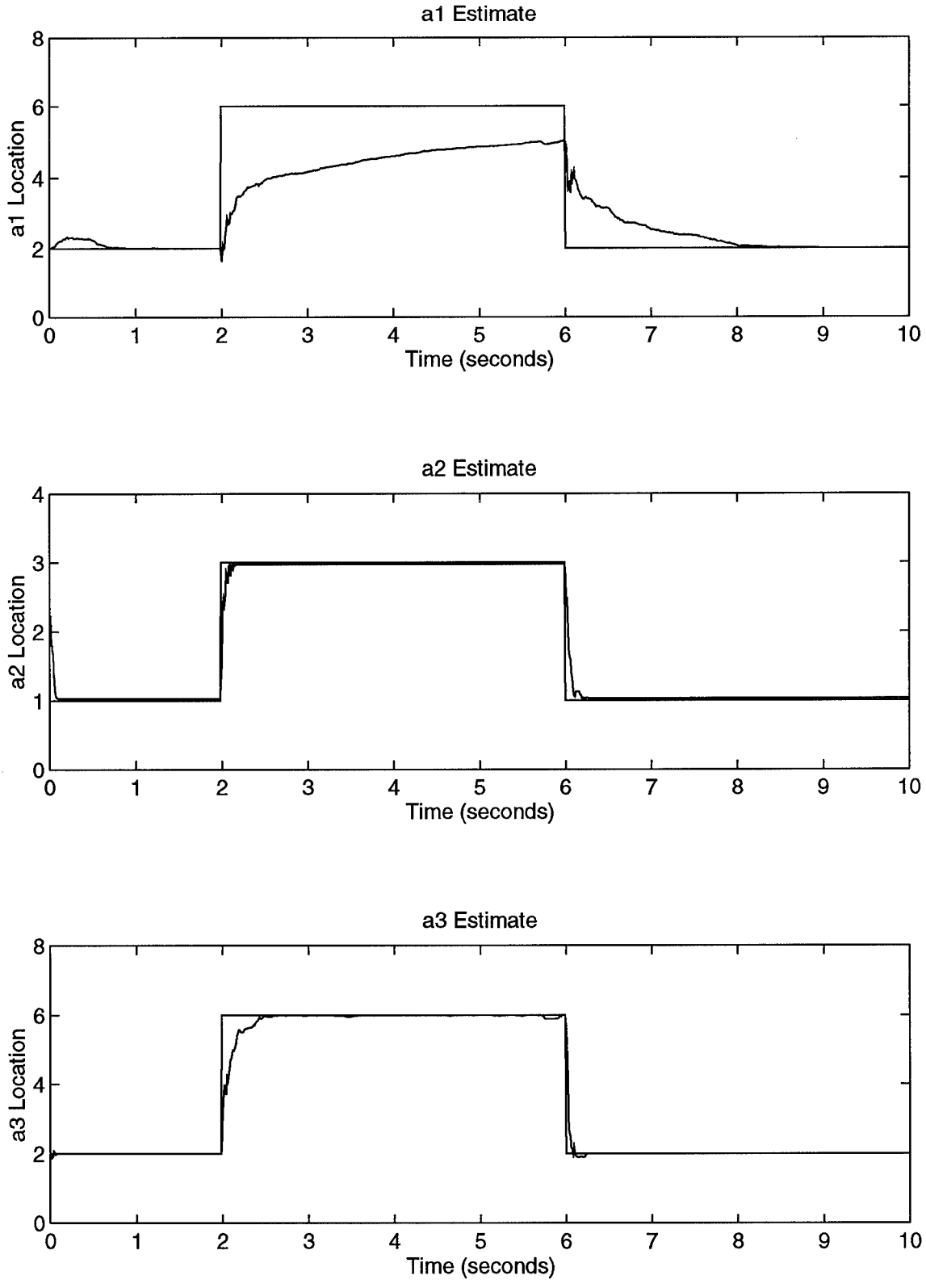


Figure A-101. 3 Parameter Jump (a_1 and a_3 Same) Using Parameter Position Mon.

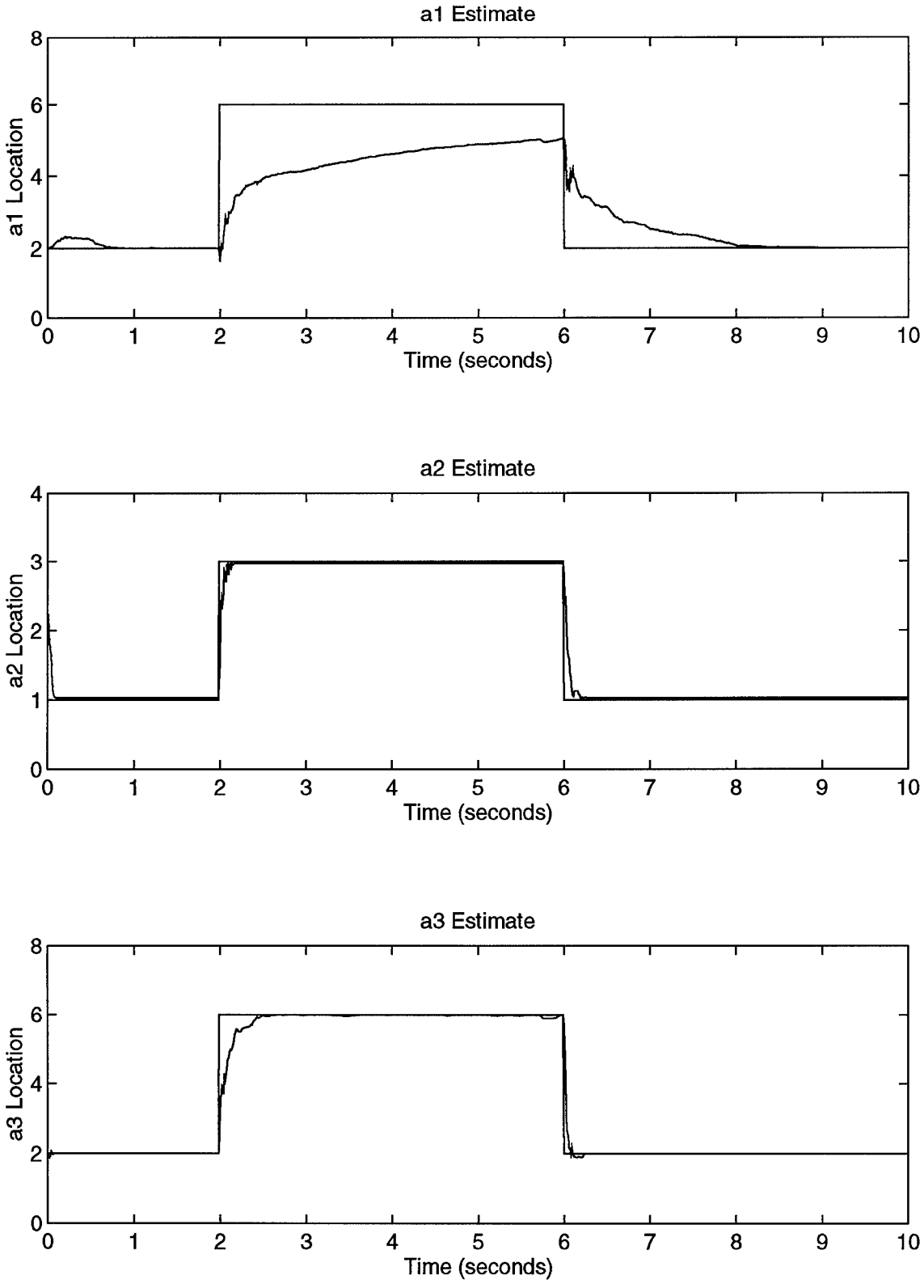


Figure A-102. 3 Parameter Jump (a_1 and a_3 Same) Using Probability Monitoring

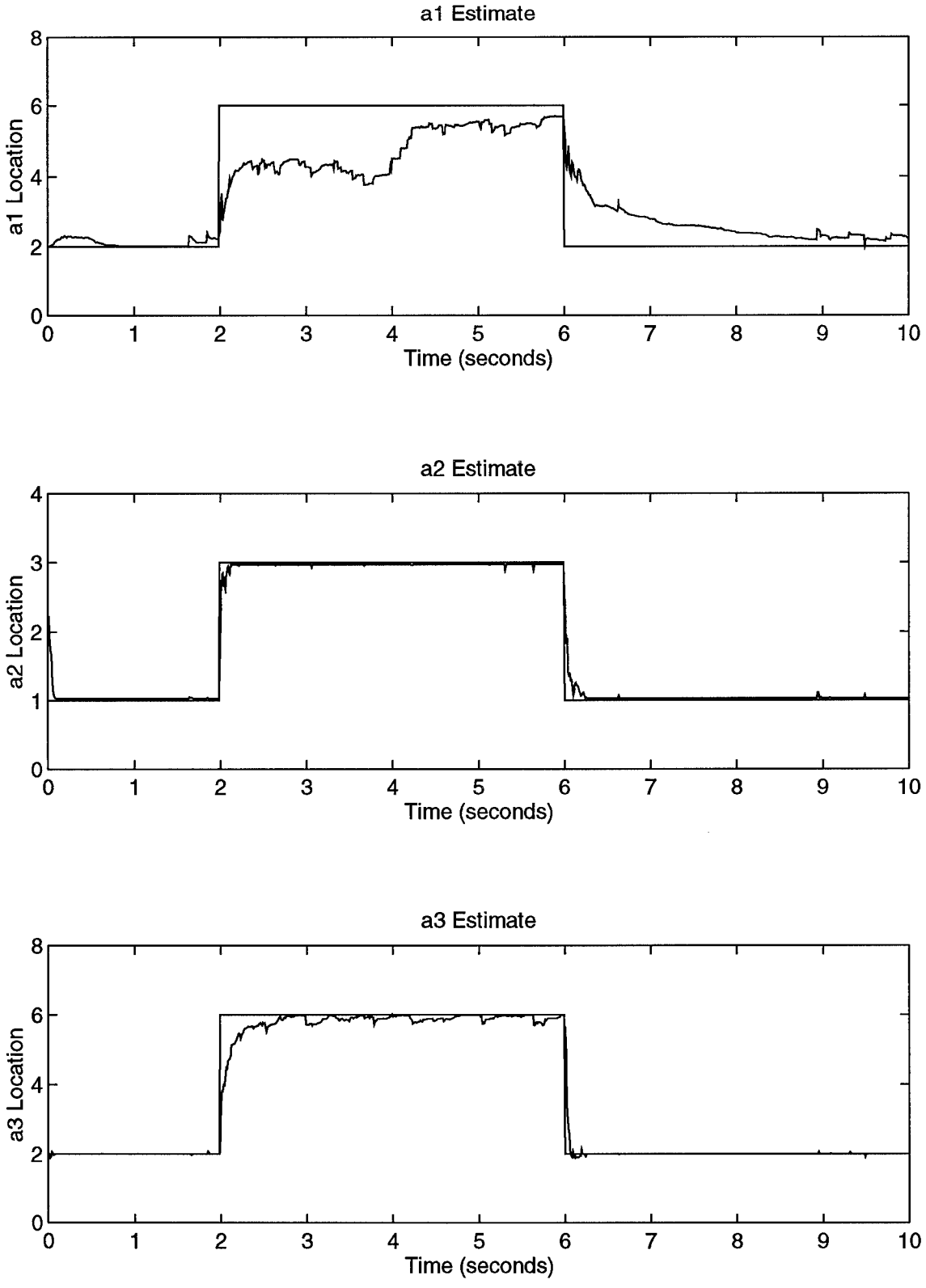
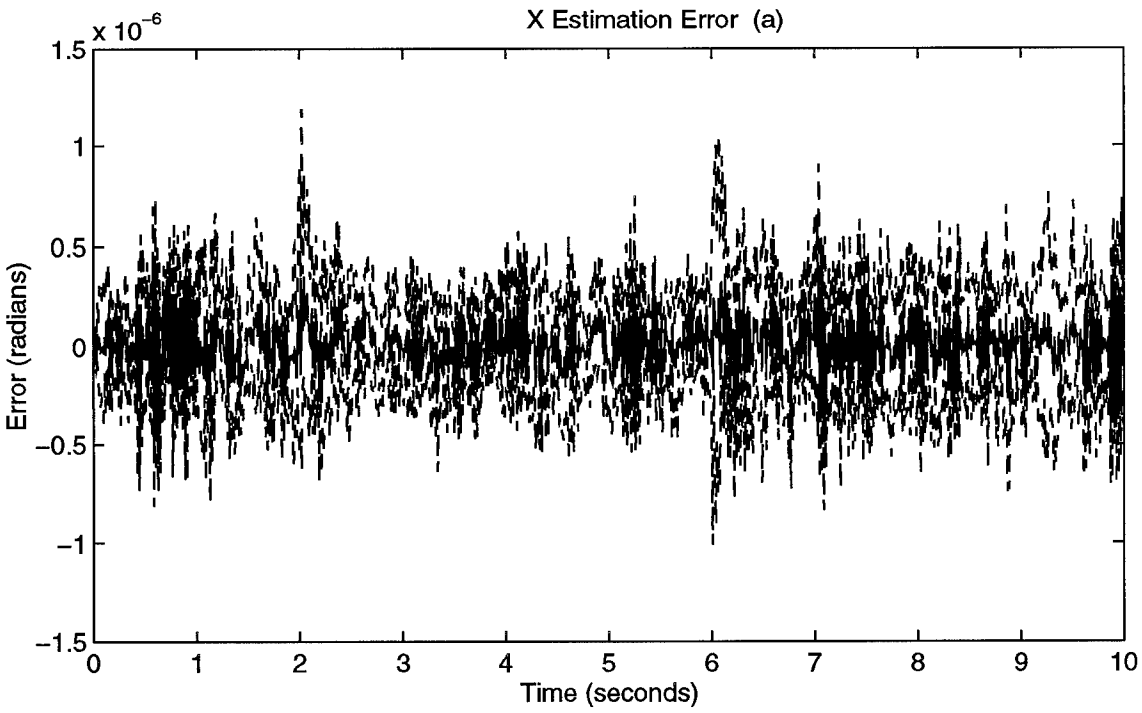


Figure A-103. 3 Parameter Jump (a_1 and a_3 Same) Using Residual Monitoring



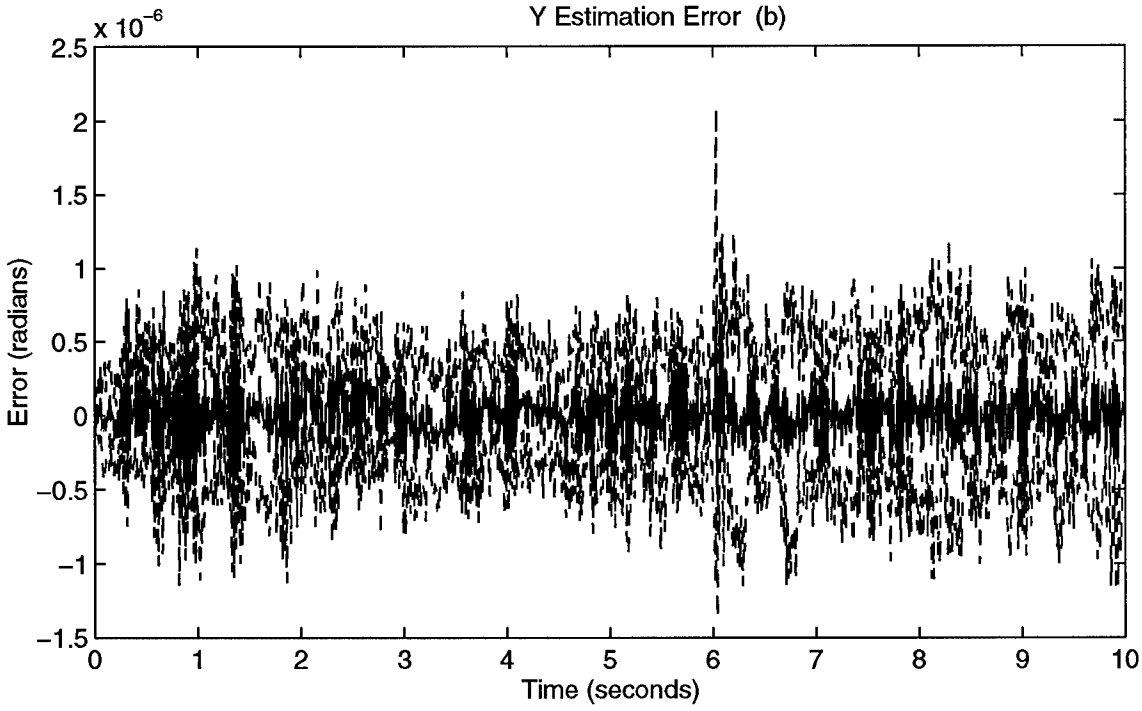
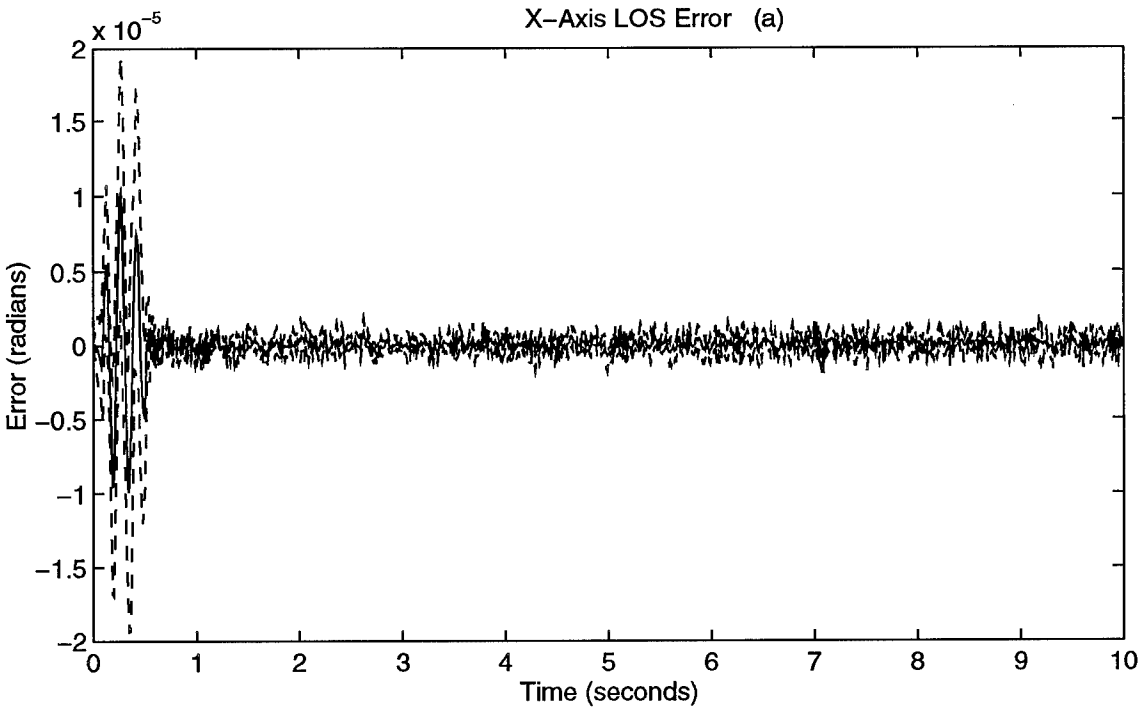


Figure A-104. 3 Parameter Jump (a_1 and a_3 Same) Estimation Errors



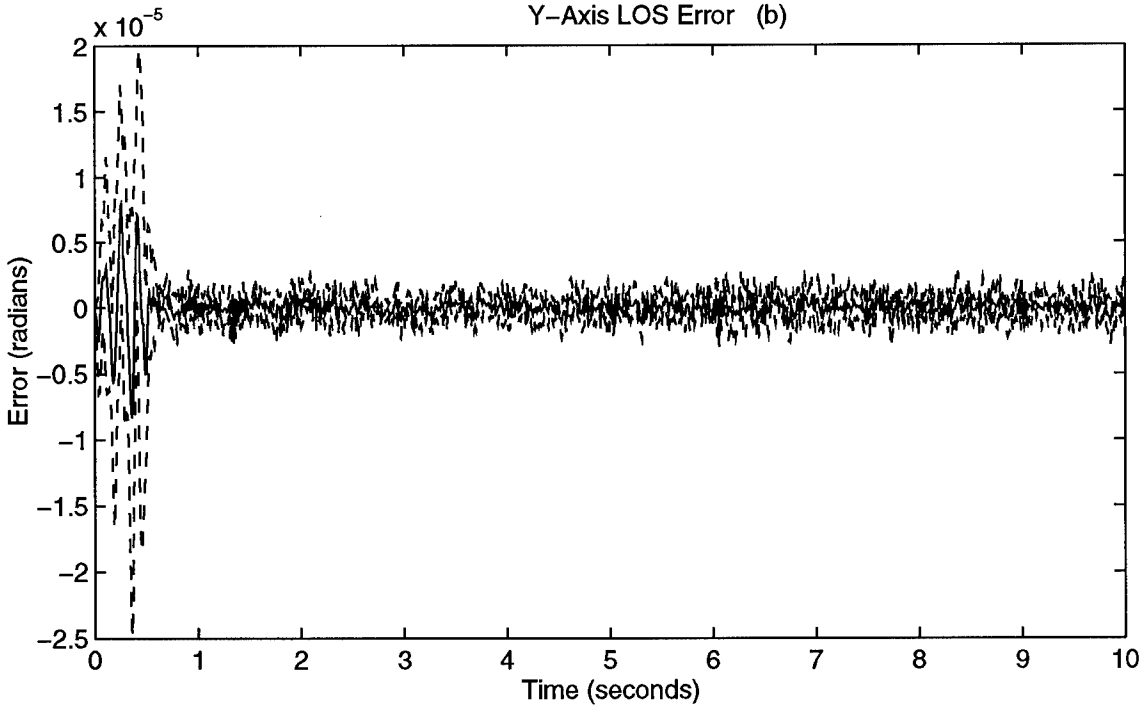


Figure A-105. 3 Parameter Jump (a_1 and a_3 Same) LOS Errors

Bibliography

1. Baram, Y., and N. R. Sandell, Jr. "An Information Theoretic Approach to Dynamic System Modeling and Identification," *IEEE Transactions on Automatic Control, AC-23 (1)*: 61-66, 1978.

2. Filios, Capt Paul G. *Moving-Bank Multiple Model Adaptive Algorithms Applied to Spacecraft Control.* MS Thesis AFIT/GE/ENG/85D-14. School of Engineering, Air Force Institute of Technology (AU), Wright-Patterson AFB, OH, Dec 1985 (AD-A164 016).

3. Fitch, Capt James. *Moving-Bank Multiple Model Adaptive Control of a Large Flexible Space Structure with Purposeful Dither for Enhanced Identifiability.* MS Thesis AFIT/GA/ENG/93D-01. School of Engineering, Air Force Institute of Technology (AU), Wright-Patterson AFB, OH, Dec 1993.

4. Fitch, Capt James. Personal interviews. School of Engineering, Air Force Institute of Technology (AU), Wright-Patterson AFB, OH, Apr through Nov 1994.

5. FORTRAN 77: IMSL Math/library: Fortran Subroutines for Mathematical Applications. IMSL, Inc., Houston, Texas, January 1989.

6. Gawronski, W. and K. B. Lim. "Controllability and Observability of Flexible Structures with Proof Mass Actuators," *Journal of Guidance, Control, and Dynamics: 16(5)*: 899-902, Sep-Oct 1993.

7. Gustafson, Capt John A. *Control of a Large Flexible Space Structure Using Multiple Model Adaptive Algorithms.* MS Thesis AFIT/GE/ENG/ 91D-22. School of Engineering, Air Force Institute of Technology (AU), Wright-Patterson AFB, OH, Dec 1991. (AD-A243 759).

8. Gustafson, J. A., and P. S. Maybeck. "Flexible Spacestructure Control via Moving-Bank Multiple Model Algorithms," *IEEE Transactions on Aerospace and Electronic Systems*, Vol. 30, No. 3: 750-757, July 1994.

9. Hentz, 1Lt Karl P. *Feasibility Analysis of Moving Bank Multiple Model Adaptive Estimation and Control Algorithms.* MS Thesis AFIT/GE/ENG/84D-32. School of Engineering, Air Force Institute of Technology (AU), Wright-Patterson AFB, OH, Dec 1984 (AD-A152 015).

10. Karnick, 2Lt Drew A. *Moving Bank Multiple Model Adaptive Estimation Applied to Flexible Spacestructure Control.* MS Thesis AFIT/GE/ENG/86D-41. School of Engineering, Air Force Institute of Technology (AU), Wright-Patterson AFB, OH, Dec 1986 (AD-A178 870).

11. Karnick, Drew A., and Peter S. Maybeck. "Moving Bank Multiple Model Adaptive Estimation Applied to Flexible Spacestructure Control," *Proceedings of the 26th IEEE Conference on Decision and Control*: 1249-1257. Los Angeles, CA, Dec 1987.

12. Kokotovic, P.V., R.E. O'Malley, Jr., and P. Sannuti. "Singular Pertubations and Order Reduction in Control Theory -- An Overview," *Automatica, 12*: 123-132, 1976.

13. Lashlee, Capt Robert W. *Moving Bank Multiple Model Adaptive Estimation Applied to Flexible Spacestructure Control.* MS Thesis AFIT/GE/ENG/87D-36. School of Engineering, Air Force Institute of Technology (AU), Wright-Patterson AFB, OH, Dec 1987.

14. Lashlee, Robert W., and Peter S. Maybeck. "Spacestructure Control Using Moving Bank Multiple Model Adaptive Estimation," *Proceedings of the 27th IEEE Conference on Decision and Control*: 712-717. Austin, TX, Dec 1988.

15. Lockheed et al. "Preliminary Design Review: Space Integrated Controls Experiment". Subtask 02-03. 21-22 Feb 1991.

16. Lockheed Missiles and Space Company, Inc. "SPICE Program, Precision Pointing Experiment, Phase I Final Report (draft)," September 1993.

17. Maghami, P. G., and S. M. Joshi. "Sensor/Actuator Placement for Flexible Space Structures," *IEEE Transactions on Aerospace and Electronic Systems, AES-29(2)*: 345-350, Apr. 1993.

18. MATRIXx: ANALYSIS AND DESIGN MANUAL. "A CAD/CAE Program." Integrated Systems Inc., Santa Clara, CA, April 1990.

19. MATLAB: Pro-MATLAB For Sun Workstations. The MathWork, Inc., South Natick, MA, January 1990.

20. Maybeck, Peter S. *Stochastic Models, Estimation, and Control, Volume 1.* NY: Academic Press, 1979.

21. Maybeck, Peter S. *Stochastic Models, Estimation, and Control, Volume 2.* NY: Academic Press, 1982.

22. Maybeck, Peter S. *Stochastic Models, Estimation, and Control, Volume 3.* NY: Academic Press, 1982.

23. Maybeck, Peter S. "Moving-Bank Multiple Model Adaptive Estimation and Control Algorithms: An Evaluation," *Control and Dynamic Systems: Advances in Aerospace System Dynamics and Control Systems, Edited by C. T. Leondes, 31*: 1-31. Academic Press, San Diego CA. 1989.

24. Maybeck, Peter S. Class handout distributed in MMAE special study session. School of Engineering, Air Force Institute of Technology (AU), Wright-Patterson AFB, OH, Dec 1992.

25. Maybeck, Peter S. Professor. Personal interviews. School of Engineering, Air Force Institute of Technology (AU), Wright-Patterson AFB, OH, Feb through Nov 1994.

26. Maybeck, Peter S., and Capt Karl P. Hentz. "Investigation of Moving-Bank Multiple Model Adaptive Algorithms," *AIAA Journal of Guidance, Control, and Dynamics*, Vol. 10, No. 1: 90-96, Jan/Feb 1987.

27. Maybeck, Peter S., and Michael R. Schore. "Reduced-Order Multiple Model Adaptive Controller for Flexible Spacestructure," *IEEE Transactions on Aerospace and Electronics Systems*, Vol. AES-28, No. 3: 756-767, July 1992.

28. Maybeck, Peter S., and R. D. Stevens. "Reconfigurable Flight Control via Multiple Model Adaptive Control Methods," *IEEE Transactions on Aerospace and Electronic Systems*, Vol. AES-27, No. 3: 470-480, May 1991.

29. Moyle, Capt Robert B. *Moving-Bank Multiple Model Adaptive Estimation and Control to a Large Space Structure.* MS Thesis AFIT/GE/ENG/90D-45. School of Engineering, Air Force Institute of Technology (AU), Wright-Patterson AFB, OH, Dec 1990.

30. Schiller, Capt Gregory. Telephone interviews. PL/XR, Kirtland AFB, NM. Mar through Nov 1994.

31. Schiller, Capt Gregory John. *Control of a Large Space Structure Using Multiple Model Adaptive Estimation and Control Techniques.* MS Thesis AFIT/GA/ENG/93D-02. School of Engineering, Air Force Institute of Technology (AU), Wright-Patterson AFB, OH, Dec 1993.

32. Schore, Capt Michael Roger. *Robustness of a Moving-Bank Multiple Model Adaptive Controller for a Large Space Structure*. MS Thesis AFIT/GE/ENG/89D-46. School of Engineering, Air Force Institute of Technology (AU), Wright-Patterson AFB, OH, Dec 1989.

33. Sheldon, Stuart N. *An Optimizing Design Strategy for Multiple Model Adaptive Estimation and Control*. PhD dissertation. School of Engineering, Air Force Institute of Technology (AU), Wright-Patterson AFB, OH, Dec 1989.

34. Skelton, Robert E, and Ajmal Yousuff. "Component Cost Analysis of Large Scale Systems"' *International Journal of Control, 37*: 285-304, 1983.

35. Van Der Werken, Capt Daniel F., Jr. *A Robustness Analysis of Moving-Bank Multiple Model Adaptive Estimation and Control of a Large Flexible Space Structure*. MS Thesis AFIT/GE/ENG/88D-59. School of Engineering, Air Force Institute of Technology (AU), Wright-Patterson AFB, OH, Dec 1988.

36. Venkayya, Vipperla B., and Victoria A. Tischler. "Frequency Control and Its Effect on the Dynamic Response of Flexible Structures," *AIAA Journal, 23*: 1768-1774, Nov 1985.

37. Williams, Trevor. "Transmission Zeros and High-Authority/Low-Authority Control of Flexible Space Structures," *AIAA Journal of Guidance, Control, and Dynamics, 17(1)*: 170-174, Jan-Feb 1994.

Vita

Captain Gordon C. Griffin, Jr. was born on 6 June 1965 in Portland, Maine. He graduated from Bourne High School in Bourne, Massachusetts in 1983 and attended Worcester Polytechnic Institute in Worcester, Massachusetts. Upon graduation in 1987 with a Bachelor of Science in Electrical Engineering, he received a reserve commission as a second lieutenant in the United States Air Force. His first assignment was at Los Angeles AFB, California, where he worked in the Program Control division of the Medium Launch Vehicle System Program Office. His main responsibilities included contractor cost analysis, cost estimation, and budget planning. In May 1990, he transferred to the engineering division of the Atlas Launch Vehicle System Program Office where he worked as an Avionics/System Safety engineer. He was responsible for the design, testing, and operational readiness of electrical power, sequencing, telemetry and tracking, and flight termination systems for Atlas and Centaur launch vehicles. He was also responsible for all vehicular safety concerns, and was a member of the launch team for five successful Atlas launches including the first ever DoD Atlas II launch. In May 1993, he entered the Graduate School of Engineering, Air Force Institute of Technology.

Permanent address: 25 Pryer Dr.
Pocasset, Massachusetts 02559

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. AGENCY USE ONLY (Leave blank)	2. REPORT DATE	3. REPORT TYPE AND DATES COVERED
	December 1994	Master's Thesis

4. TITLE AND SUBTITLE
CONTROL OF A LARGE SPACE STRUCTURE USING MULTIPLE MODEL ADAPTIVE ESTIMATION AND CONTROL TECHNIQUES

5. FUNDING NUMBERS

6. AUTHOR(S)
Gordon C. Griffin, Jr.

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
Air Force Institute of Technology, WPAFB OH 45433-6583

8. PERFORMING ORGANIZATION REPORT NUMBER
AFIT/GE/ENG/94D-14

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)
Phillips Laboratory
PL/LITT
Kirtland AFB, NM 87117-5776

10. SPONSORING/MONITORING AGENCY REPORT NUMBER

11. SUPPLEMENTARY NOTES

12a. DISTRIBUTION/AVAILABILITY STATEMENT
Distribution Unlimited

12b. DISTRIBUTION CODE

13. ABSTRACT *(Maximum 200 words)*

ABSTRACT

A moving bank multiple model adaptive estimator/controller (MMAE/MMAC), based on linear system, quadratic cost, and Gaussian noise (LQG) assumptions, is used to quell unwanted vibrations in a large flexible space structure. The structure, known as the SPace Integrated Controls Experiment, or SPICE, exists at Phillips Laboratory, Kirtland AFB, NM. The structure consists of a large platform and a smaller platform connected by a tripod of flexible legs. The purpose of the control system is to maintain a very precise line-of-sight vector through the center of the spacecraft. Kalman filtering, used to estimate the position and velocity of the bending modes of the structure, and LQG control techniques are the primary design tools used in the MMAE/MMAC algorithms. Implementing a parallel bank of filters increases robustness when uncertainties exist in the system model, here specifically allowing adaptation to uncertain and changing undamped natural frequencies of the bending modes of the structure. A moving bank is used to reduce computational loading. The MMAE/MMAC design method provides an excellent method of estimating undamped natural frequency variations and quelling vibrations in the structure. The MMAE/MMAC was able to track numerous parameter changes and jumps while providing adequate control for the structure.

14. SUBJECT TERMS
Multiple Model Adaptive Estimation, Multiple Model Adaptive Control, LQG Control, Kalman Filter, Flexible Space Structure

15. NUMBER OF PAGES
247

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
UNCLASSIFIED	UNCLASSIFIED	UNCLASSIFIED	UL

GENERAL INSTRUCTIONS FOR COMPLETING SF 298

The Report Documentation Page (RDP) is used in announcing and cataloging reports. It is important that this information be consistent with the rest of the report, particularly the cover and title page. Instructions for filling in each block of the form follow. It is important to ***stay within the lines*** to meet ***optical scanning requirements***.

Block 1. Agency Use Only *(Leave blank)*.

Block 2. Report Date. Full publication date including day, month, and year, if available (e.g. 1 Jan 88). Must cite at least the year.

Block 3. Type of Report and Dates Covered. State whether report is interim, final, etc. If applicable, enter inclusive report dates (e.g. 10 Jun 87 - 30 Jun 88).

Block 4. Title and Subtitle. A title is taken from the part of the report that provides the most meaningful and complete information. When a report is prepared in more than one volume, repeat the primary title, add volume number, and include subtitle for the specific volume. On classified documents enter the title classification in parentheses.

Block 5. Funding Numbers. To include contract and grant numbers; may include program element number(s), project number(s), task number(s), and work unit number(s). Use the following labels:

C	- Contract	**PR**	- Project
G	- Grant	**TA**	- Task
PE	- Program Element	**WU**	- Work Unit Accession No.

Block 6. Author(s). Name(s) of person(s) responsible for writing the report, performing the research, or credited with the content of the report. If editor or compiler, this should follow the name(s).

Block 7. Performing Organization Name(s) and Address(es). Self-explanatory.

Block 8. Performing Organization Report Number. Enter the unique alphanumeric report number(s) assigned by the organization performing the report.

Block 9. Sponsoring/Monitoring Agency Name(s) and Address(es). Self-explanatory.

Block 10. Sponsoring/Monitoring Agency Report Number. *(If known)*

Block 11. Supplementary Notes. Enter information not included elsewhere such as: Prepared in cooperation with...; Trans. of...; To be published in.... When a report is revised, include a statement whether the new report supersedes or supplements the older report.

Block 12a. Distribution/Availability Statement. Denotes public availability or limitations. Cite any availability to the public. Enter additional limitations or special markings in all capitals (e.g. NOFORN, REL, ITAR).

- **DOD** - See DoDD 5230.24, "Distribution Statements on Technical Documents."
- **DOE** - See authorities.
- **NASA** - See Handbook NHB 2200.2.
- **NTIS** - Leave blank.

Block 12b. Distribution Code.

- **DOD** - Leave blank.
- **DOE** - Enter DOE distribution categories from the Standard Distribution for Unclassified Scientific and Technical Reports.
- **NASA** - Leave blank.
- **NTIS** - Leave blank.

Block 13. Abstract. Include a brief *(Maximum 200 words)* factual summary of the most significant information contained in the report.

Block 14. Subject Terms. Keywords or phrases identifying major subjects in the report.

Block 15. Number of Pages. Enter the total number of pages.

Block 16. Price Code. Enter appropriate price code *(NTIS only)*.

Blocks 17. - 19. Security Classifications. Self-explanatory. Enter U.S. Security Classification in accordance with U.S. Security Regulations (i.e., UNCLASSIFIED). If form contains classified information, stamp classification on the top and bottom of the page.

Block 20. Limitation of Abstract. This block must be completed to assign a limitation to the abstract. Enter either UL (unlimited) or SAR (same as report). An entry in this block is necessary if the abstract is to be limited. If blank, the abstract is assumed to be unlimited.

★U.S.GPO:1990-0-273-271